

Міністерство освіти і науки України
Донбаська державна машинобудівна академія

**ПРАКТИЧНІ РОБОТИ
З ДИСЦИПЛІНИ
«ПРОГРЕСИВНІ МЕХАТРОННІ СИСТЕМИ
МАШИНОБУДІВНОГО ВИРОБНИЦТВА»**

Методичні вказівки

**для здобувачів третього освітньо-наукового рівня вищої освіти
спеціальності 133 «Галузеве машинобудування»,
денної та заочної форм навчання**

Краматорськ – Тернопіль
ДДМА
2023

УДК 621

Практичні роботи з дисципліни «Прогресивні мехатронні системи машинобудівного виробництва» *методичні вказівки [для здобувачів третього освітньо-наукового рівня вищої освіти спеціальності 133 «Галузеве машинобудування», денної та заочної форм навчання]* / [уклад.: В.Д. Ковальов]. – Краматорськ; Тернопіль : ДДМА, 2023. – 58 с.

Практичні роботи з дисципліни «Прогресивні мехатронні системи машинобудівного виробництва» містять порядок виконання лабораторних і практичних робіт. Наведено короткі теоретичні відомості, деякі довідкові дані, зміст звіту, завдання, контрольні питання.

Для здобувачів третього освітньо-наукового рівня вищої освіти спеціальності 133 «Галузеве машинобудування».

Укладач:

В.Д. Ковальов, проф.

Відп. за випуск

Я. В. Васильченко, проф.

ЗМІСТ

ЛАБОРАТОРНА РОБОТА №1. ВИВЧЕННЯ ДАТЧИКІВ ПОЛОЖЕННЯ...	4
ЛАБОРАТОРНА РОБОТА № 2. ВИВЧЕННЯ ДИНАМІЧНИХ ХАРАКТЕРИСТИК КРОКОВОГО ПРИВОДУ	12
ЛАБОРАТОРНА РОБОТА № 3. ВИВЧЕННЯ СЕРЕДОВИЩА РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ МІКРОКОНТРОЛЕРІВ СІМЕЙСИВА AVR	18
ПРАКТИЧНА РОБОТА № 1. ВИВЧЕННЯ АРХІТЕКТУРИ ТА СИСТЕМИ КОМАНД МІКРОКОНТРОЛЕРІВ СІМЕЙСТВА AVR.....	23
ПРАКТИЧНА РОБОТА № 2. РОБОТА З ПОРТАМИ ВВЕДЕННЯ- ВИВЕДЕННЯ, ПРОГРАМУВАННЯ ПРОСТИХ ЛІНІЙНИХ І РОЗГАЛУЖЕНИХ АЛГОРИТМІВ.	32
ПРАКТИЧНА РОБОТА № 3. ВИВЧЕННЯ МОЖЛИВОСТЕЙ ВБУДОВАНИХ ТАЙМЕРІВ-ЛІЧИЛЬНИКІВ.	39
ПРАКТИЧНА РОБОТА № 4. ВИВЧЕННЯ СИСТЕМИ ПЕРЕРИВАНЬ. ..	49
Завдання до лабораторної роботи.....	57
ЛІТЕРАТУРА.....	59

ЛАБОРАТОРНА РОБОТА №1 ВИВЧЕННЯ ДАТЧИКІВ ПОЛОЖЕННЯ

Мета роботи: вивчити основні типи датчиків положення, принципи їх роботи. Ознайомитися зі схемами включення і досліджувати основні характеристики датчиків.

Прилади та обладнання

1. Фото імпульсний датчик типу ВЕ-178А5.
2. Кінцеві вимикачі різних типів.
3. Обертовий трансформатор ВТМ 1В.
4. Безконтактний індуктивний датчик БВК400.
5. Безконтактний індуктивний датчик КВД-3-24.
6. Індикатор годинникового типу.
7. Набір УСП.
8. Джерело живлення 24В.
9. Індикаторна лампа.

Основні теоретичні відомості

1. *Фотоімпульсні датчики* – фотоелектричні (оптичні) вимірювальні перетворювачі дозволяють забезпечити високу стабільність відліку і добре узгоджуються з цифровими електронними пристроями, зокрема з системами ЧПУ, в яких вони отримали переважне застосування.

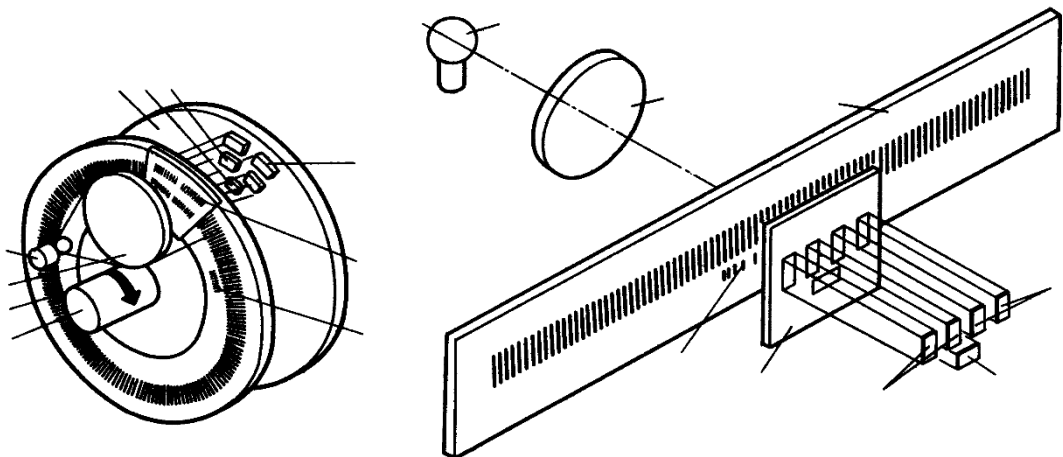


Рисунок 1 – Круговий (а) і лінійний (б) оптичний імпульсний ІІІ

Конструкції фото імпульсних датчиків кругового і лінійного типів показані на (рис. 1). Основа датчика – оптична пара, що складається з джерела світла 1 і фотоприймачів 7, 8, 9, між якими знаходяться шкала 3 і знімач 6. Шкала кругового датчика розміщується на валу 4, а інші елементи – нерухомо на корпусі 10. Шкала і знімач датчика являють собою скляні пластини, на які нанесені непрозорі штрихи, причому ширина

штрихів дорівнює ширині проміжків між ними. Світло від джерела 1 (лампочка або світлодіод) збирається лінзою 2 і прямує крізь знімач і шкалу. Якщо штрихи на шкалі і на знімачі збігаються, то світло проходить крізь прозорі проміжки між штрихами. При зміщенні шкали на половину кроку штрихів щодо знімача, штрихи на знімачі виявляються навпроти проміжків на шкалі, і світло не проходить до фотоприймача. На знімачі штрихи розташовані в чотири сектори, зміщені один щодо одного на $1/4$ кроку штрихів. Навпаки кожного з секторів знімача розташоване по фотодіоду, які видають первинні сигнали, що змінюються при зміщенні шкали щодо знімача по синусоїдальному і косинусоїдальному закону (рис. 1, а). Крім цих основних сигналів в більшості датчиків виробляється ще й сигнал нульової мітки, створений окремою групою штрихів 5 і окремою парою фотодіодів.

Первинні сигнали всіх шести фотодіодів надходять на різномірні входи диференціальних (балансних) підсилювачів, які підсилюють різницю двох сигналів. Причому до одного підсилювача підключається пара фотодіодів, для яких штрихи на знімачі зміщені на півкроку, тобто для сигналу синуса – перший і третій фотодіоди, а для косинуса – другий і четвертий. Цим досягається компенсація постійної складової сигналу, багаторазово підвищується завадостійкість і знижується залежність сигналу від якості шкали, знімача і інтенсивності світла від джерела. Потім посилені сигнали і сигнал початку відліку подаються на входи формувачів, що перетворюють синусоїдальну напругу в прямокутні імпульси. Амплітуда цих імпульсів не залежить у певних межах від зміни амплітуди первинних синусоїдальних сигналів, а періоди синусоїдального і прямокутного сигналів рівні. Підрахунок числа імпульсів помноженого на крок штрихів дає величину вимірюваного переміщення, а застосування двох сигналів рахунку, зрушених відносно один одного на $1/4$ кроку, тобто синусоїдального і косинусоїдального, дозволяє при подальшій обробці визначати напрямок переміщення.

Серійно випускаються кругові датчики мають кількість штрихів на шкалі від 50 до 5000, спеціальні прецизійні-до 200 000. Лінійні датчики випускаються з кроком штрихів: 40, 20, 4 і 2 мкм.

2. *Кінцеві та шляхові вимикачі* є приладами контролю стану об'єкта під впливом керуючих упорів шляхом механічного контакту в певних точках шляху. Вони знаходять застосування як елемент контролю та управління технологічними процесами у всіх галузях промисловості.

По конструкції вимикачі відрізняються видом перемикаючого пристрою, яке може бути прямого (без гістерезису) і полумиттєвого (з гістерезисом) дії, з одним замикаючим і одним контактом, що розмикає (рис. 2). Крім того, вони розрізняються по виду чутливого елемента (рис. 3): а – кнопковий, б – роликівий поступальний, в – роликівий поворотний.

ковальсько-пресовому обладнанні та інших виробничих механізмах, де потрібна висока надійність і довговічність.

Найбільшого поширення набули індуктивні вимикачі, що використовують електромагнітне поле. В основі їх роботи покладено принцип керованого релаксаційного генератора. Структурна схема такого датчика показана на рис. 4.

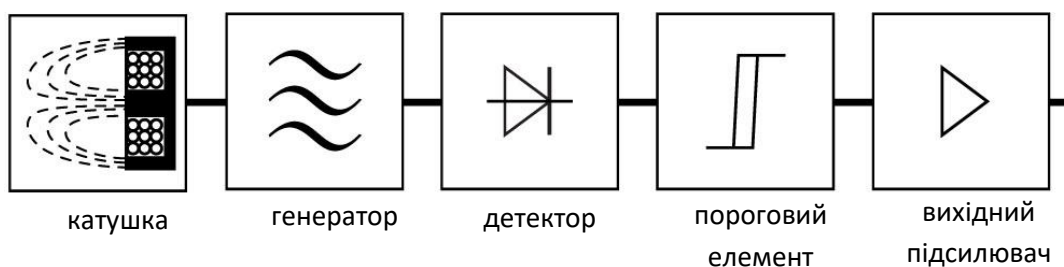


Рисунок 4 – Структурна схема індуктивного кінцевого вимикача

Управління генератором здійснюється введенням в полі котушки індуктивності металевої пластини, яка змінює настройки котушки. При введенні пластини в щілину датчика в ній наводяться вихрові струми віднімають частину потужності коливань генератора. При цьому амплітуда цих коливань знижується. При виведенні пластини з щілини добротність котушки зростає, і відповідно збільшується амплітуда коливань. Ці коливання перетворюються детектором в постійну напругу, яка подається на пороговий елемент (тригер Шмідта). Перемикання стану цього елемента відбувається при строго певній вхідній напрузі. Наявність гістерезису в пороговому елементі підвищує стійкість датчика до перешкод. Вихідний підсилювач підсилює вихідний сигнал до рівня, необхідного для підключення зовнішнього виконавчого органу, наприклад електромагнітного реле.

Спрацьовування вимикача серії БВК-400 відбувається при введенні в робочий зазор алюмінієвої пластини товщиною не менше 1,5 мм і шириною в не менше 11 мм (рис. 5). Відстань між двома рухомими сусідніми пластинами, що викликають спрацьовування одного і того ж вимикача, не менше 9 мм. Відстань (m) від нижньої кромки пластини до основи щілини від 1 до 4 мм. Спрацьовування вимикача відбувається при положенні перемикає пластини за віссю Р котушки на відстані $D_0 = 0...3$ мм в залежності від зразка.

При русі пластини зверху положення перемикає пластини, при якому відбувається спрацьовування вимикача, лежить за віссю Q на відстані 14 мм від неї.

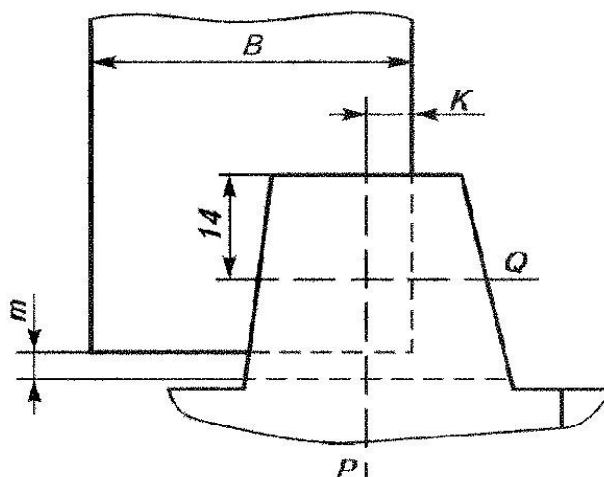


Рисунок 5 – Положення перемикає пластини в момент спрацьовування вимикачів серії БВК-400

На мал. 6 приведена схема приєднання вимикача до джерела живлення і підключення до нього реле Р.

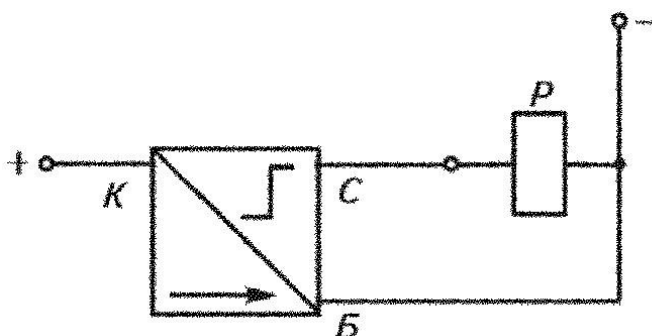


Рисунок 6 – Схема включення вимикача типу БВК-400

Технічні характеристики датчиків типу БВК-400

Номінальна напруга живлення УН0М, В.....	24
Допустиме коливання напруги живлення	(0,85-1,24) УН0М
Опір навантаження не менше, Ом	120
Розкид довжини шляху спрацьовування при коливанні напруги живлення (0,85-1,24)УН0М, мм, не більше	10,1
Максимальний розкид довжини шляху спрацьовування при зміні температури від -10 до + 45°С, мм, не більше.....	+0,5
Диференціал ходу (гістерезис), мм, не більше	3
Максимальна частота спрацьовування, кГц, не нижче	1
Потужність, комутована вимикачем без урахування потужності, споживаної навантаженням, Вт, не більше	0,4

4. *Обертвий трансформатор (резольвер)* – це мікромашина змінного струму, що служить для перетворення кута повороту ротора в синусоїдальну напругу з фазою пропорційної кутку повороту ротора.

Статор обертового трансформатора (рис. 7) має дві однофазні обмотки С1-С2 і С3-С4, зсунуті на 90° відносно один одного. Ротор також має дві обмотки, включені послідовно (на малюнку для спрощення показана одна обмотка Р1-Р2). Магнітопровід виготовляють з листів електротехнічної сталі або пермаллоя. На роторі і статорі рівномірно розташовані пази, в яких розміщені взаємно перпендикулярні обмотки. Первинними обмотками обертового трансформатора (ВТ) найчастіше є обмотки статора, а вторинними – обмотки ротора, який пов'язаний з нерухомих елементом. При повороті ротора, який власне і є первинним датчиком, з нього знімається синусоїдальна напруга.

Струмознімання з обмоток ротора відбувається за допомогою контактних кілець і щіток. Статорні обмотки ВТ харчуються змінною напругою з частотою 400, 1000, 2000 або 4000 Гц. Закон зміни напруги живлення повинен бути синусоїдальним і косинусоїдальним з коефіцієнтом вищих гармонік не більше 0,05%. При з'єднанні обмоток статора до мережі змінного струму утворюється обертове магнітне поле. Стрілка на рис. 6 показує результуючий магнітний потік, що проходить через ротор в даний момент часу. Магнітне поле обертається зі швидкістю, що відповідає частоті живлення обмоток статора.

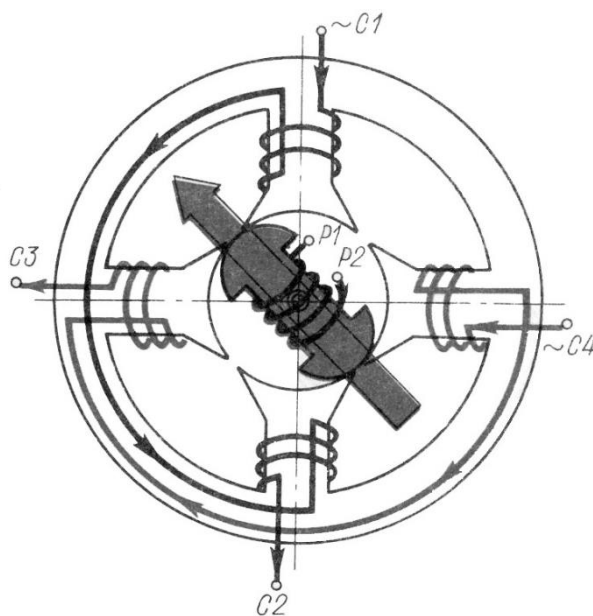


Рисунок 6 – Схема роботи обертового трансформатора

Обертове магнітне поле індукуює в обмотці ротора змінну напругу, амплітуда якого постійна, а зсув фази щодо напруги живлення фази

залежить від положення ротора. Якщо ротор розташований вертикально, то фази вихідного і вхідного сигналів збігаються. При повороті ротора на кут α фаза вихідного сигналу зміниться на кут в k раз більше кута α , де k – число пар полюсів датчика. Таким чином, при повороті вала ротора механічний кут обертання перетворюється в електричний, тобто зсув фаз.

Широке застосування знаходять багатополюсні ВТ. Принцип дії десятиполюсного ВТ такий самий, як і двополюсного. Різниця цих двох ВТ полягає в електричному зсуві фаз щодо механічного кута обертання. Точність вимірювання кута повороту за допомогою обертового трансформатора становить зазвичай $0,1 \dots 0,01$.

Порядок виконання роботи

1. За наданими зразками датчиків вивчити їх конструкцію і принцип дії.
2. З комплекту УСП і індикатора годинникового типу, за вказівками лаборанта зібрати пристосування для контролю параметрів кінцевих вимикачів.
3. Встановити на пристосування механічний кінцевий вимикач з кнопковим штовхачем напівмиттєвої дії, і підключити до джерела живлення індикаторну лампу через замикають контакти кінцевого вимикача.
4. Переміщаючи за допомогою гвинта штовхач вимикача, домогтися включення лампи і зафіксувати показання індикатора. Потім, обертаючи гвинт в зворотну сторону домогтися виключення лампи і знову зафіксувати показання. Дослід повторити 10 разів.
5. За результатами вимірювань оцінити величину гістерезису вимикача і точність спрацювання. Для цього в кожному експерименті розрахувати середнє значення з десяти дослідів для координати включення і для координати виключення. Різниця між середньою координатою включення і середньою координатою виключення дасть величину гістерезису, а величина дисперсії по кожному параметру – охарактеризує точність спрацювання.
6. Аналогічні експерименти провести з механічним кінцевим вимикачем прямої дії і з безконтактним кінцевим вимикачем, підключивши його за схемою, показаної на рис. 6.

Зміст звіту

1. Назва роботи, мета, прилади та обладнання.
2. Схему контрольного стенду.
3. Таблиці з результатами вимірювань для трьох датчиків.
4. Результати розрахунків величини гістерезису і точності спрацювання для трьох датчиків.
5. Висновок.

Контрольні питання

1. Область застосування кругових і лінійних оптичних датчиків шляху.
2. Принцип роботи кругового оптичного імпульсного датчика.
3. Принцип роботи лінійного оптичного імпульсного датчика.
4. Назвіть переваги і недоліки оптичних імпульсних датчиків?
5. Назвіть основні типи механічних кінцевих вимикачів, область їх застосування, переваги і недоліки.
6. Індуктивні кінцеві вимикачі. Принцип роботи, переваги і недоліки.
7. Обертається трансформатор, область застосування, принцип роботи.

ЛАБОРАТОРНА РОБОТА № 2 ВИВЧЕННЯ ДИНАМІЧНИХ ХАРАКТЕРИСТИК КРОКОВОГО ПРИВОДУ

Мета роботи: вивчити залежність крутного моменту крокового двигуна від швидкості обертання, визначити основні динамічні показники крокового двигуна.

Прилади та обладнання.

1. Кроковий двигун типу ДШР-40-0012.
2. Блок управління кроковим двигуном.
3. Персональний комп'ютер.
4. Програма управління кроковим приводом.
5. Навантажувальний пристрій.
6. Набір вантажів.
7. Маховик.

Опис лабораторної установки.

Схема лабораторної установки показана на рис. 2.1.

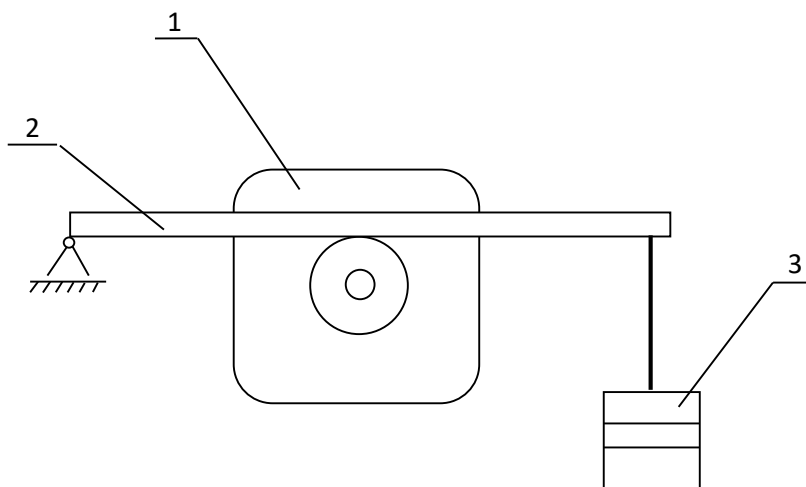


Рисунок 2.1 – Схема лабораторної установки.

Лабораторна установка містить два крокових двигуна, один з яких навантажується статично гальмівним пристроєм, наведених на рисунку, де 1 – двигун, 2 – важіль з гальмівною колодкою, 3 – набір вантажів; другий навантажений динамічно маховиком (на схемі не показаний). Зміна статичного навантаження проводиться шляхом зміни кількості вантажів, що призводить до зміни сили тертя. Наявність або відсутність втрати кроку оцінюється за характером обертання гальмівного шків: рівномірне без ривків-нормальний режим, наявність Ривків – втрата кроку. Аналогічно оцінюється режим руху двигуна з маховиком.

Основні теоретичні відомості

Кроковий двигун – це електромеханічний пристрій, що перетворюють сигнал управління в кутове (або лінійне) переміщення ротора з фіксацією його в заданому положенні без пристроїв зворотного зв'язку. Найбільш універсальною різновидом крокових двигунів являються так звані гібридні або редукторні. Гібридні двигуни забезпечують малу величину кроку, більший момент і більшу швидкість в порівнянні з іншими видами. Типове число кроків на оборот для гібридних двигунів становить від 100 до 400 (кут кроку 3.6– 0.9). Гібридні двигуни поєднують в собі кращі риси двигунів зі змінним магнітним опором і двигунів з постійними магнітами. Ротор гібридного двигуна має зубці, розташовані в осьовому напрямку (рис. 2.2).

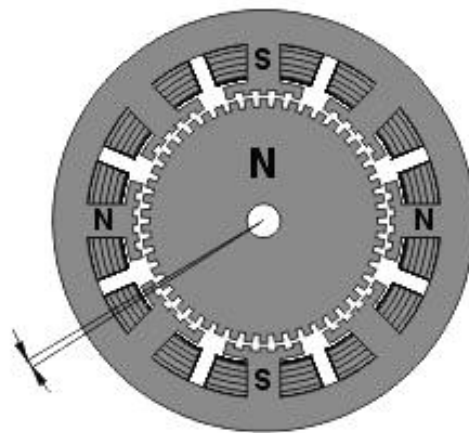


Рисунок 2.2 – Поперечний розріз гібридного крокового двигуна

Ротор розділений на дві частини, між якими розташований циліндричний постійний магніт. Таким чином, зубці передньої половинки ротора є північними полюсами, а зубці задньої половинки – південними. Крім того, половинки ротора повернені один щодо одного на половину кута кроку зубців. Число пар полюсів ротора дорівнює кількості зубців на одній з його половинок. Зубчасті полюсні наконечники ротора, як і статор, набрані з окремих пластин для зменшення втрат на вихрові струми. Статор гібридного двигуна також має зубці, забезпечуючи велику кількість еквівалентних полюсів, на відміну від основних полюсів, на яких розташовані обмотки. Зазвичай використовуються 4 основних полюси для 3.6 двигунів і 8 основних полюсів для 1.8 з'єднувальних і 0.9 двигунів. Зубці ротора забезпечують менший опір магнітної ланцюга в певних положеннях ротора, що покращує статичний і динамічний момент. Це забезпечується відповідним розташуванням зубців, коли частина зубців ротора знаходиться строго навпроти зубців статора, а частина між ними.

Залежність між числом полюсів ротора, числом еквівалентних полюсів статора і числом фаз визначає кут кроку S двигуна:

$$S = \frac{360}{N_{ph} \cdot P_h} = \frac{360}{N}$$

де, N_{ph} – число еквівалентних полюсів на фазу, одно числу полюсів ротора;

P_h – число фаз;

N – повна кількість полюсів для всіх фаз разом.

Ротор показаного на малюнку двигуна має 100 полюсів (50 пар), двигун має 2 фази, тому повне кількість полюсів – 200, а крок, відповідно, 1.8° .

Подовжній перетин гібридного крокового двигуна показано на рис. 2.3. Стрілками показано напрямок магнітного потоку постійного магніту ротора. Частина потоку (на малюнку показана чорною лінією) проходить через полюсні наконечники ротора, повітряні зазори і полюсний наконечник статора. Ця частина не бере участі у створенні моменту.

Як видно на малюнку, повітряні зазори у верхнього і нижнього полюсного наконечника ротора різні. Це досягається завдяки повороту полюсних наконечників на половину кроку зубів. Тому існує інша магнітна ланцюг, яка містить мінімальні повітряні зазори і, як наслідок, володіє мінімальним магнітним опором. По цьому ланцюгу замикається інша частина потоку (на малюнку показана штриховою білою лінією), яка і створює момент. Частина ланцюга лежить в площині, перпендикулярної малюнку, тому не показана. У цій же площині створюють магнітний потік котушки статора. У гібридному двигуні цей потік частково замикається полюсними наконечниками ротора, і постійний магніт його «бачить» слабо. Тому на відміну від двигунів постійного струму, магніт гібридного двигуна неможливо розмагнітити ні при якій величині струму обмоток.

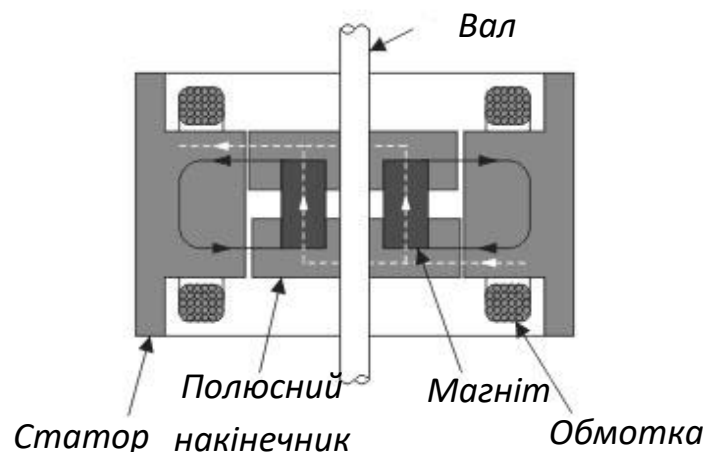


Рисунок 2.3 – Подовжній розріз гібридного крокового двигуна

Такий параметр крокового двигуна, як залежність моменту від швидкості є найважливішим при виборі типу двигуна, метод керування фазами і схеми драйвера. При конструюванні високошвидкісних драйверів крокових двигунів потрібно враховувати, що обмотки двигуна являють собою індуктивність. Ця індуктивність визначає час наростання і спаду струму. Тому якщо до обмотці прикладено напругу прямокутної форми, форма струму не буде прямокутною. При низьких швидкостях (рис. 4, а) час наростання і спаду струму не здатне сильно вплинути на момент, однак на високих швидкостях момент падає. Пов'язано це з тим, що на високих швидкостях струм в обмотках двигуна не встигає досягти номінального значення (рис. 2.4, б).

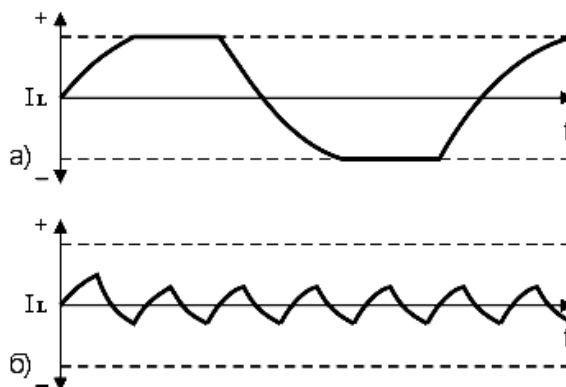


Рисунок 2.4 - Форма струму в обмотках двигуна на різних швидкостях роботи

Для того щоб момент падав як можна менше, необхідно забезпечити високу швидкість наростання струму в обмотках двигуна, що досягається застосуванням спеціальних схем для їх харчування.

Поведінка моменту при збільшенні частоти комутації фаз приблизно таке: починаючи з деякої частоти зрізу момент монотонно падає. Зазвичай для крокового двигуна наводяться дві криві залежності моменту від швидкості (рис. 5).

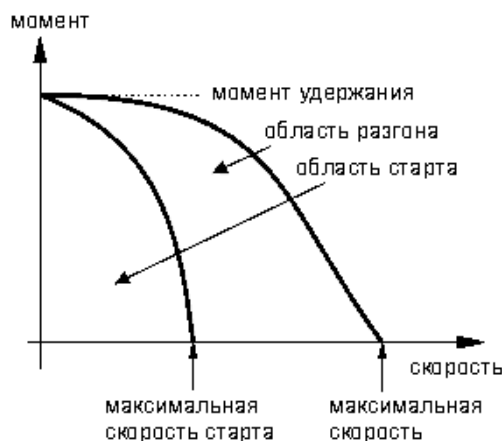


Рисунок 2.5 – Залежність крутного моменту від швидкості

Внутрішня крива (крива старту) показує, при якому максимальному моменті навантаження для даної швидкості кроковий двигун здатний рушити з місця. Ця крива перетинає вісь швидкостей в точці, званої максимальною частотою старту або частотою прийомистості. Вона визначає максимальну швидкість, на якій ненавантажений двигун може рушити. На практиці ця величина лежить в межах 200 – 500 повних кроків в секунду. Інерційність навантаження сильно впливає на вигляд внутрішньої кривої. Велика інерційність відповідає меншій області під кривою. Ця область називається областю старту. Зовнішня крива (крива розгону) показує, при якому максимальному моменті тертя для даної швидкості кроковий двигун здатний підтримувати обертання без пропуску кроків. Ця крива перетинає вісь швидкостей в точці, званої максимальною частотою розгону. Вона показує максимальну швидкість для даного двигуна без навантаження. При вимірюванні максимальної швидкості потрібно мати на увазі, що через явище резонансу момент дорівнює нулю ще й на резонансній частоті. Область, яка лежить між кривими, називається областю розгону.

Опис програми управління кроковим приводом.

Програма керування кроковим приводом знаходиться у файлі Step_mtr.exe і працює під управлінням MS-DOS чи Windows-9x. Управління програмою здійснюється з клавіатури з допомогою клавіш Page Up, Page Down, цифрових і функціональних клавіш.

Клавіші мають наступне призначення:

F1 – пуск двигуна з поточною швидкістю;

F2 – зупинка двигуна;

F3 – введення довільного значення швидкості (тільки в режимі зупинки);

Page Up – збільшення поточної швидкості на одиницю;

Page Down – зменшення поточної швидкості на одиницю;

Порядок виконання роботи

1. Включити персональний комп'ютер і блок управління.
2. Завантажити програму управління кроковим приводом.
3. Шляхом поступового збільшення швидкості обертання визначити максимальну швидкість обертання без навантаження. Результат записати в звіт.
4. Шляхом періодичного включення і виключення двигуна, кожен раз змінюючи швидкість, визначити максимальну швидкість старту без навантаження, використовуючи метод половинного поділу. Результат записати в звіт.

5. Для заданих значень навантаження, шляхом поступового збільшення швидкості, визначити максимальну швидкість обертання під навантаженням. Результати вимірювань занести в таблицю 1.

6. За даними таблиці 1 побудувати графік залежності крутного моменту від швидкості обертання, зробити висновки по виконаній роботі.

Таблиця 2.1

нагрузка	частота	нагрузка	частота
1		9	
2		10	
3		11	
4		12	
5		13	
6		14	
7		15	
8		16	

Зміст звіту.

1. Тема.
2. Мета роботи.
3. Схема лабораторної установки.
4. Таблиця з результатами змін.
5. Графік залежності крутного моменту від швидкості.
6. Список основних динамічних параметрів двигуна.
7. Висновок.

Контрольні питання

1. Як залежить крутний момент крокового двигуна від швидкості обертання?
2. Поясніть фізичну природу цієї залежності.
3. Яким чином можна збільшити крутний момент на великій швидкості обертання?
4. Перерахуйте відомі вам типи крокових двигунів.
5. Перерахуйте основні переваги і недоліки крокового приводу.

ЛАБОРАТОРНА РОБОТА № 3

ВИВЧЕННЯ СЕРЕДОВИЩА РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ МІКРОКОНТРОЛЕРІВ СІМЕЙСИВА AVR

Мета роботи: вивчити порядок роботи і основні можливості програмного пакету AVR Studio.

Основні теоретичні відомості.

AVR Studio - професійна інтегроване середовище розробки програмного забезпечення, призначена для написання і налагодження прикладних програм для AVR мікропроцесорів в середовищі Windows 9x/NT/2000/XP. AVR Studio містить компілятор мови асемблер, компілятор мови з і віртуальний програмний симулятор для перевірки і налагодження розробляється програмного забезпечення. Крім того AVR Studio підтримує апаратні засоби налагодження програм на реальному пристрої і програматори наступних моделей: ICE50, ICE40, JTAGICE, ICE200, STK500/501/502 і AVRISP.

Після запуску **AVR Studio**, відкриється вікно створення проекту, показане на рис 1. Для створення нового проекту вибираємо мову програмування (**Atmel AVR Assembler** або **AVR GCC – СІ компілятор**), назва (**Project name**) та директорію (**Location**), де буде створено новий проект, після чого натискаємо кнопку «**Next**».



Рисунок 3.1 – Вікно створення проекту

Після цього у вікні (рис. 2) вибираємо налагоджувальну платформу **Debug platform** (для роботи з віртуальним симулятором вибираємо **AVR Simulator**) і модель мікроконтролера **Device**, для якого буде розроблятися

програмне забезпечення, і натискаємо кнопку Finish. Коли всі ці дії виконані, з'являється робоча область AVR Studio (рис. 3.2).

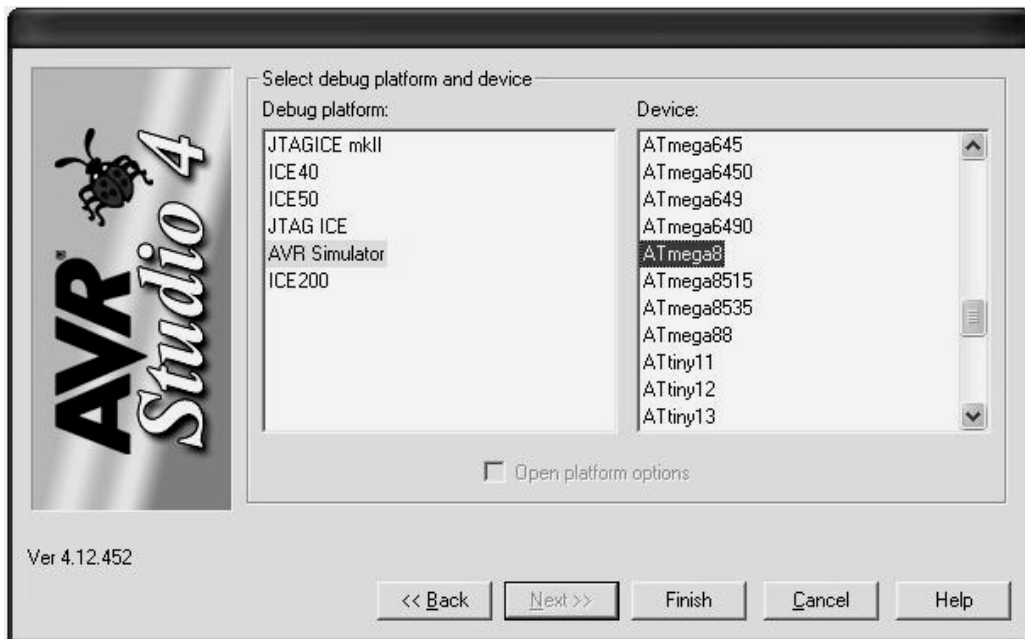


Рисунок 3.2 – Вікно вибору платформи і моделі контролера.

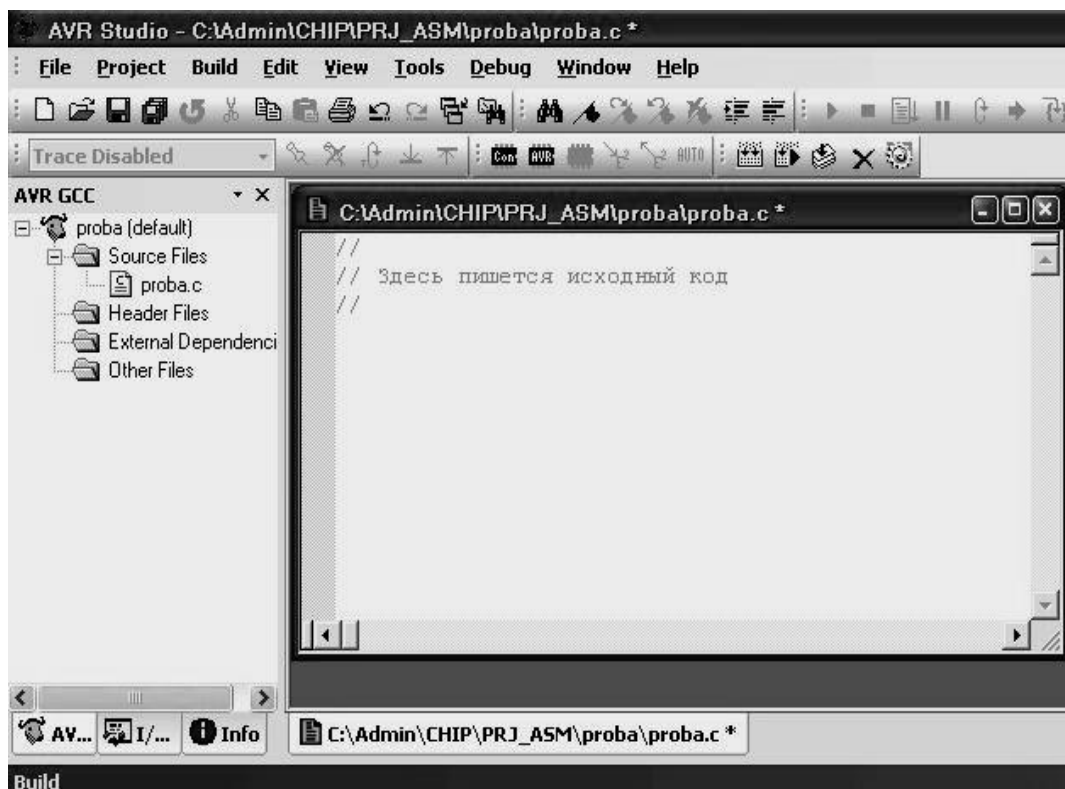


Рисунок 3.3 – Робоча область AVR Studio

Робоча область містить наступні елементи: 1 – головне меню – містить в текстовій формі повний набір команд і налаштувань AVR Studio, частина з яких винесена на панелі керування; 2 – панель «стандартна» - містить загальноприйняті команди роботи з файлами, буфер обміну, вивід

тексту з'явиться вказівник виконання (жирна жовта стрілка), який спочатку встановлений на першу команду програми.

4. В даному режимі доступні наступні команди і варіанти виконання програми (див. кнопки на панелі відладчика з лева на право):

- запуск налагодження (Run);
- вихід з режиму налагодження;
- запуск програми на виконання повністю;
- пауза (останов виконання запущеної програми в довільний момент часу);
- рестарт програми (обнуляє всі регістри процесора і встановлює покажчик на початок програми);
- перейти на покажчик (виводить на екран ту частину тексту програми, в якій в даний момент знаходиться курсор виконання);
- покрокове виконання програми зі входом в підпрограми (після кожного натискання виконується одна команда);
- покрокове виконання програми без входу в підпрограми (кожна підпрограма виконується як одна команда);
- вихід з підпрограми (при знаходженні покажчика всередині підпрограми натискання цієї кнопки призводить до автоматичної доопрацювання решти підпрограми про зупинку після виходу в основну програму);
- швидкий підхід до підпрограми (при знаходженні покажчика в основній програмі натискання цієї кнопки призводить до автоматичного виконання команд поки не зустрінеться виклик підпрограми);
- автоматичний покроковий режим (імітує періодичне натискання на кнопку покрокового виконання);
- вставити точку зупину (при автоматичному виконанні програми відбувається зупинка на команді, яка відзначена точкою зупинки).

Найбільш вживаними для початківця програміста режимами виконання є режими покрокової відпрацювання. Вони дозволяють оцінювати стан всіх елементів контролера, використовуючи дерево периферії в лівому вікні, після виконання кожної команди програми, і таким чином найбільш ефективно виявляти і виправляти помилки. При налагодженні великих програм або програм з циклами більш зручним інструментом є точки зупину. Однак їх використання вимагає від програміста деякого досвіду і чіткого знання структури налагоджують програми.

Порядок виконання роботи

1. Запустити AVR Studio.
2. Створити новий проект.

3. Вивчити набір команд головного меню, розташування основних вікон і елементів управління.
4. Скопіювати у вікно вихідного тексту текст програми з набору прикладів.
5. Виконати компіляцію програми.
6. Відпрацювати програму в покроковому режимі, спостерігаючи за зміною стану елементів контролера.
7. Зробити висновки і оформити звіт.

Зміст звіту

1. Тема і мета роботи.
2. Перелік основних команд з меню Build, Edit, View, Tools, Debug з розшифровкою їх значення.
3. Перелік додаткових вікон, що відкриваються з меню View.
4. Висновки з описом основних можливостей програмного пакету AVR Studio.

ПРАКТИЧНА РОБОТА № 1

ВИВЧЕННЯ АРХІТЕКТУРИ ТА СИСТЕМИ КОМАНД МІКРОКОНТРОЛЕРІВ СІМЕЙСТВА AVR

Мета роботи: вивчити архітектуру, принцип роботи AVR-мікроконтролерів, групи команд, їх синтаксис, правила використання команд різних видів.

Основні теоретичні відомості.

Мікроконтролер або однокристальна мікро-ЕОМ являє собою універсальне функціонально закінчене програмоване обчислювальний пристрій, призначене для створення на його базі пристроїв управління малої і середньої складності. Сучасні мікроконтролери виготовляються, як правило, у вигляді однієї мікросхеми і містять: процесор, тактовий генератор, оперативну та постійну пам'ять, а також різний набір периферійних пристроїв, таких як таймери-лічильники, порти вводу-виводу, аналого-цифрові перетворювачі, аналогові компаратори, послідовні синхронні і асинхронні прийомопередавачі з різними протоколами обміну і т. п..

На рис. 1 представлена структурна схема мікроконтролера типу ATtiny2313, одного з найбільш простих представників мікро контролерів сімейства AVR. Основу мікроконтролера становить процесор, що містить арифметико-логічний пристрій, пристрій керування і файл регістрів загального призначення. Цей регістровий файл містить 32 восьмирозрядних регістра загального призначення, доступ до яких здійснюється за один машинний цикл. Тому за один машинний цикл виконується одна операція АЛУ. Два операнда вибираються з реєстрового файлу, виконується операція, результат її записується в реєстровий файл-все за один машинний цикл.

Шість з 32 регістрів можна використовувати як три 16-розрядних покажчика в адресному просторі даних, що дає можливість використовувати високоефективну адресну арифметику (16-розрядні регістри X, Y і Z). Один з трьох адресних покажчиків (регістр Z) можна використовувати для адресації таблиць в пам'яті програм.

АЛУ підтримує арифметичні та логічні операції регістр-регістр, і константа – регістр. Операції над окремими регістрами також виконуються в АЛУ.

Крім реєстрових операцій, для роботи з регістровим файлом можуть використовуватися доступні режими адресації, оскільки регістровий файл займає адреси \$00-\$1F в області даних, звертатися до них як до комірок пам'яті. Простір вводу складається з 64 адрес для периферійних пристроїв процесора, таких як керуючі регістри, таймери-лічильники та інші. Доступ

до простору вводу-виводу може здійснюватися безпосередньо, як до комірок пам'яті розташованим після реєстрового файлу (\$20-\$5F).

Процесори AVR побудовані за Гарвардською архітектурою з роздільними областями пам'яті програм і даних. Доступ до пам'яті програм здійснюється за допомогою однорівневого буфера. Під час виконання чергової команди, наступна команда вибирається з пам'яті програм. Подібна концепція дає можливість виконувати по одній команді за кожен машинний цикл. Пам'ять програм-це внутрішньо системна завантажувальна флеш-пам'ять.

За допомогою команд відносних переходів і виклику підпрограм здійснюється доступ до всього адресного простору. Велика частина команд AVR має розмір 16-розрядів. Кожна адреса в пам'яті програм містить одну 16-розрядну команду.

При обробці переривань і виклику підпрограм адреса повернення запам'ятовується в стеку. Стек розміщується в пам'яті даних загального призначення, відповідно розмір стека обмежений тільки розміром доступної пам'яті даних і її використанням в програмі. Всі програми Користувача повинні ініціалізувати покажчик стека (SP) в програмі виконуваної після скидання (до того як викликаються підпрограми і дозволяються переривання). 8-розрядний покажчик стека доступний для читання/запису в області вводу/виводу.

Доступ до 128 байтів статичного ОЗУ, реєстрового файлу і реєстрів введення / виводу здійснюється за допомогою п'яти доступних режимів адресації підтримуваних архітектурою AVR. Весь простір пам'яті AVR є лінійним і безперервним.

Гнучкий модуль переривань має власний керуючий реєстр в просторі вводу/виводу, і прапор глобального дозволу переривань в реєстрі стану. Кожному перериванню призначений свій вектор в початковій області пам'яті програм. Різні переривання мають пріоритет відповідно до розташування їх векторів. За молодшими адресами розташовані вектори з великим пріоритетом.

Всі команди оперують реєстрами прямо адресуються до будь-якого з реєстрів за один машинний цикл. Єдиний виняток-п'ять команд оперують з константами SBCI, SUBI, CPI, ANDI, ORI і команда LDI, що завантажує реєстр константою. Ці команди працюють тільки з другою половиною реєстрового файлу-R16...R31. Команди SBC, SUB, CP, AND I OR, також як і всі інші, застосовні до всього реєстрового файлу.

Кожному реєстру присвоєно адресу в просторі даних, вони відображаються на перші 32 комірки ОЗУ. Хоча реєстровий файл фізично розміщений поза ОЗП, подібна організація пам'яті дає гнучкий доступ до

регістрів. Регістри X, Y і Z можуть використовуватися для індексації будь-якого регістру.

АЛУ процесора безпосередньо підключено до 32 регістрів загального призначення. За один машинний цикл АЛУ проводить операції між регістрами реєстрового файлу. Команди АЛУ розділені на чотири основні категорії: арифметичні і логічні, розгалуження, пересилання і бітові. Їх опис представлено в таблицях 1 – 4.

Таблиця 1 - Арифметичні та логічні команди

Мнемонік а	Операнд и	Опис	Дія	Прапори	Цикл
ADD	Rd, Rr	Скласти два регістри	$Rd \leftarrow Rd + Rr$	Z,C,N,V,H	1
ADC	Rd, Rr	Скласти з перенесення м	$Rd \leftarrow Rd + Rr + C$	C,Z,C,N,V,H	1
ADIW	Rdi, K	Скласти слово з константою	$Rdh,l \leftarrow Rdh,l + K$	Z,C,N,V,S	2
SUB	Rd, K	Відняти два регістри	$Rd \leftarrow Rd - Rr$	Z,C,N,V,H	1
SUBI	Rd, K	Відняти константу	$Rd \leftarrow Rd - K$	Z,C,N,V,H	1
SBIW	RdI, K	Відняти з слова константу	$Rdh,l \leftarrow Rdh,l - K$	Z,C,N,V,S	2
SBC	Rd, Rr	Відняти з перенесення м	$Rd \leftarrow Rd - Rr - C$	Z,C,N,V,H	1
SBCI	Rd, K	Відняти з перенесення м	$Rd \leftarrow Rd - K - C$	Z,C,N,V,H	1
And	Rd, R	Логічне та	$Rd \leftarrow Rd \text{ AND } Rr$	Z,N,V	1
ANDI	Rd,K	Логічне та з константою	$Rd \leftarrow Rd \text{ AND } K$	Z,N,V	1
OR	Rd,RR	Логічне або	$Rd \leftarrow Rd \text{ OR } Rr$	Z,N,V	1
ORI	Rd,K	Логічне або зконстантою	$Rd \leftarrow Rd \text{ OR } K$	Z,N,V	1
EOR	Rd, Rr	Виключає або	$Rd \leftarrow Rd \text{ XOR } Rr$	Z,N,V	1
COM	Rd	Доповнення до	$Rd \leftarrow \$FF - Rd$	Z,C,N,V	1
NEG	Rd	Доповнення до	$Rd \leftarrow \$00 - Rd$	Z,C,N,V,H	1
SBR	Rd, K	Вставити біт(i) в регістрі	$Rd \leftarrow Rd \text{ OR } K$	Z,N,V	1
CBR	Rd, K	Скинути	$Rd \leftarrow Rd \text{ AND } (FFh)$	Z,N,V	1

		біт(i) в реєстрі	-K)		
INC	Rd	Збільшити на 1	Rd<Rd+1	Z,N,V	1
DEC	Rd	Зменшити на 1	Rd<Rd-1	Z,N,V	1
TST	Rd	Перевірити на 0 або мінус	Rd<Rd AND Rd	Z,N,V	1
CLR	Rd	Очистити реєстр	Rd<Rd XOR Rd	Z,N,V	1
SER	Rd	Встановити реєстр	Rd<\$FF	–	1

Таблиця 2 – Команди розгалуження

Мнемоніка	Операнди	Опис	Дія	Прапори	Цикл
RJMP	K	Відносний перехід	PC<PC+k+1	–	2
LJMP		Перехід за адресою(Z)	PC<Z	–	2
RCALL	K	Віднесення виклик підпрог.	PC<PC+k+1	–	3
ICALL		Виклик підп.за адр.(Z)	PC<Z	–	3
RET		Вихід з підпрог.	PC<STACK	–	4
RETI		Вихід з перериванням	PC<STACK	I	4
CPSE	Rd, Rr	Порівняти,пропуск якщо рівне	if(Rd=Rr) PC<PC+2 или 3	–	1/2
CP	Rd, Rr	Порівнювати	Rd-Rr	Z,N,V,C,H	1
CPC	Rd, Rr	Порівняти з перенесенням	Rd-Rr-C	Z,N,V,C,H	1
CPI	Rd, K	Порівняти з константою	Rd-K	Z,N,V,C,H	1
SBRC	Rr, b	Пропуск якщо біт в реєстрі скинутий	if(Rr(b)=0) PC<PC+2 или 3	–	1/2
SBRS	Rr, b	Пропуск якщо біт в реєстрі встановлений	if(Rr(b)=1) PC<PC+2 или 3	–	1/2
SBIC	P, b	Пропуск якщо в рег.введення/вив скинутий	if(P(b)=0) PC<PC+2 или 3	–	1/2
SBIS	P, b	Пропуск якщо біт в рег.введення/вив установлений	if(P(b)=1) PC<PC+2 или 3	–	1/2
BRBS	s, k	Перехід якщо встановлений прапор s	if(SREG(s)=1) PC<PC+k+1	–	1/2

BRBC	s, k	Перехід якщо скинутий прапор s	if(SREG(s)=0) PC<PC+k+1	–	1/2
BREQ	K	Перехід якщо рівно	if(Z=1) PC<PC+k+1	–	1/2
BRNE	K	Перехід якщо нерівно	if(Z=0) PC<PC+k+1	–	1/2
BRCS	K	Перехід якщо встановлено перенесення	if(C=1) PC<PC+k+1	–	1/2
BRCC	K	Перехід якщо скинуто перенесення	if(C=0) PC<PC+k+1	–	1/2
BRSH	K	Перехід якщо дорівнює або більше	if(C=0) PC<PC+k+1	–	1/2
BRLO	K	Перехід, якщо менше	if(C=1) PC<PC+k+1	–	1/2
BRMI	K	Перехід якщо мінус	if(N=1) PC<PC+k+1	–	1/2
BRPL	K	Перехід якщо плюс	if(N=0) PC<PC+k+1	–	1/2
BRGE	K	Перехід, якщо більше або дорівнює, зі знаком	if(N XOR V=0) PC<PC+k+1	–	1/2
BRLT	K	Перехід якщо менше нуля, зі знаком	if(N XOR V=1) PC<PC+k+1	–	1/2
BRHS	K	Перехід якщо встановлено прапор H	if (H=1) PC<PC+k+1	–	1/2
BRHC	K	Перехід якщо скинутий прапор H	if (H=0) PC<PC+k+1	–	1/2
BRTS	K	Перехід якщо встановлено прапор T	if (H=1) PC<PC+k+1	–	1/2
BRTC	K	Перехід якщо скинутий прапор T	if (H=0) PC<PC+k+1	–	1/2
BRVS	K	Перехід якщо встановлений прапор V	if (H=1) PC<PC+k+1	–	1/2
BRVC	K	Перехід якщо скинутий прапор V	if (H=0) PC<PC+k+1	–	1/2
BRIE	K	Перехід якщо дозволені переривання	if(I=1) PC<PC+k+1	–	1/2
BRID	K	Перехід якщо заборонені переривання	if(I=1) PC<PC+k+1	–	1/2

Таблиця 3 - Команди пересилання

Мнемоніка	Операнди	Опис	Дія	Прапори	Цикл
MOV	Rd, Rr	Пересилання між регістрами	Rd<Rr	–	1
LDI	Rd, K	Завантажити константу	Rd<K	–	1
LD	Rd, X	Завантажити регістр безпосередньо	Rd<(X)	–	2
LD	Rd, X+	Завантажити регістр безпосередньо з подальшим інкрементом	Rd<(X), X<X+1	–	2
LD	Rd, -X	Завантажити регістр безпосередньо з попереднім декрементом	X<X-1, Rd<(X)	–	2
LD	Rd, Y	Завантажити регістр безпосередньо	Rd<(Y)	–	2
LD	Rd, Y+	Завантажити регістр безпосередньо з подальшим інкрементом	Rd<(Y), Y<Y+1	–	2
LD	Rd, -Y	Завантажити регістр безпосередньо з попереднім декрементом	Y<Y-1, Rd<(Y)	–	2
LDD	Rd, Y+q	Завантажити регістр безпосередньо зі зміщенням	Rd<(Y+q)	–	2
LD	Rd, Z	Завантажити регістр безпосередньо	Rd<(Z)	–	2
LD	Rd, Z+	Завантажити регістр безпосередньо з подальшим інкрементом	Rd<(Z), Z<Z+1	–	2
LD	Rd, -Z	Завантажити регістр безпосередньо з попереднім декрементом	Z<Z-1, Rd<(Z)	–	2
LDD	Rd, Z+q	Завантажити регістр безпосередньо зі зміщенням	Rd<(Z+q)	–	2
LDS	Rd, k	Завантажити з	Rd<(k)	–	3

		ОЗУ			
ST	X, Rr	Записати реєстр безпосередньо	$(X) < Rr$	–	2
ST	X+, Rr	Записати реєстр безпосередньо з подальшим інкрементом	$(X) < Rr,$ $X < X+1$	–	2
ST	-X, Rr	Записати реєстр безпосередньо з попереднім декрементом	$X < X-1,$ $(X) < Rr$	–	2
ST	Y, Rr	Записати реєстр безпосередньо	$(Y) < Rr$	–	2
ST	Y+, Rr	Записати реєстр безпосередньо з подальшим інкрементом	$(Y) < Rr,$ $Y < Y+1$	–	2
ST	-Y, Rr	Записати реєстр безпосередньо з попереднім декрементом	$Y < Y-1,$ $(Y) < Rr$	–	2
STD	Y+q, Rr	Записати реєстр безпосередньо зі зміщенням	$(Y+q) < Rr$	–	2
ST	Z, Rr	Записати реєстр безпосередньо	$(Z) < Rr$	–	2
ST	Z+, Rr	Записати реєстр безпосередньо з подальшим інкрементом	$(Z) < Rr,$ $Z < Z+1$	–	2
ST	-Z, Rr	Записати реєстр безпосередньо з попереднім декрементом	$Z < Z-1,$ $(Z) < Rr$	–	2
STD	Y+q, Rr	Записати реєстр безпосередньо зі зміщенням	$(Y+q) < Rr$	–	2
STS	k, Rr	Записати в ОЗУ	$(k) < Rr$	–	3
LPM	–	Завантажити R0 з пам'яті програм	$R0 < (Z)$	–	3
IN	Rd, P	Введення з порту	$Rd < P$	–	1
OUT	P, Rr	Виведення в	$P < Rr$	–	1

		порт			
PUSH	Rr	Записати в стек	STACK<Rr	–	2
POP	Rr	Прочитати з стека	Rr<STACK	–	2

Таблиця 4 – Команди роботи з бітами

Мнемоніка	Операнди	Опис	Дія	Прапори	Цикл
SBI	P, b	Встановити біт в регістрі введення / виводу	I/O(P,b)<1	–	2
CBI	P, b	Скинути біт в регістрі введення / виводу	I/O(P,b)<0	–	2
LSL	Rd	Логічний зсув вліво	Rd(n+1)<Rd(n), Rd(0)<0	Z,C,N,V	1
LSR	Rd	Логічний зсув вправо	Rd(n)<Rd(n+1), Rd(7)<0	Z,C,N,V	1
ROL	Rd	Циклічний зсув вліво через C	Rd(0)<C, Rd(n+1)<Rd(n), C<Rd(7)	Z,C,N,V	1
ROR	Rd	Циклічний зсув вправо через C	Rd(7)<C, Rd(n)<Rd(n+1), C<Rd(0)	Z,C,N,V	1
ASR	Rd	Арифметичний зсув вправо	Rd(n)<Rd(n+1), n=0..6	Z,C,N,V	1
SWAP	Rd	Обмін нібблов	Rd(3-0)<Rd(7-4), Rd(7-4)<Rd(3-0)	–	1
BSET	S	Встановити прапор в регістрі стану	SREG(s)<1	SREG(s)	1
BCLR	S	Скинути прапор в регістрі стану	SREG(s)<0	SREG(s)	1
BST	Rr, b	Запам'ятати біт в T	T<Rr(b)	T	1
BLD	Rd, b	Прочитати біт з T	Rd(b)<T	–	1
SEC	–	Встановити перенесення	C<1	C	1
CLC	–	Скинути перенесення	C<0	C	1
SEN	–	Встановити прапор N	N<1	N	1
CLN	–	Скинути ПРАПОР N	N<0	N	1
SEZ	–	Встановити прапор Z	Z<1	Z	1
CLZ	–	Скинути	Z<0	Z	1

		прапор Z			
SEI	–	Дозволити переривання	I<1	I	1
CLI	–	Заборонити переривання	I<0	I	1
SES	–	Встановити прапор S	S<1	S	1
CLS	–	Скинути прапор S	S<0	S	1
SEV	–	Встановити прапор V	V<1	V	1
CLV	–	Скинути прапор V	V<0	V	1
SET	–	Встановити прапор T	T<1	T	1
CLT	–	Скинути ПРАПОР T	T<0	T	1
SEH	–	Встановити прапор H	H<1	H	1
CLH	–	Скинути прапор H	H<0	H	1
NOP	–	Немає операції	–	–	1
SLEEP	–	Зупин	–	–	3
WDR	–	Скидання сторожового таймера	–	–	1

Порядок виконання роботи

1. Запустити AVR Studio.
2. Створити новий проект.
3. Вивчити набір команд.
4. Шляхом введення і виконання найпростіших програм в режимі симуляції випробувати дії основних команд.
5. Зробити висновки і оформити звіт.

Зміст звіту

1. Тема і мета роботи.
2. Структурна схема мікроконтролера.
3. Тексти тестових програм для випробування десяти команд за вказівкою керівника.
4. Висновки з описом основних можливостей мікроконтролера.

Контрольні питання

1. Які групи команд ви знаєте?
2. Що спільного між командами однієї групи?
3. Які види адресації ви знаєте?
4. Призначення регістрів-показчиків.
5. Переваги та недоліки архітектури з роздільною пам'яттю програм і даних.

ПРАКТИЧНА РОБОТА № 2

РОБОТА З ПОРТАМИ ВВЕДЕННЯ-ВИВЕДЕННЯ, ПРОГРАМУВАННЯ ПРОСТИХ ЛІНІЙНИХ І РОЗГАЛУЖЕНИХ АЛГОРИТМІВ.

Мета роботи: вивчити порядок роботи з портами введення-виведення мікроконтролерів сімейства AVR. Отримати практичні навички написання простих прикладних програм з лінійними і розгалуженими алгоритмами.

Прилади та обладнання:

1. Персональний комп'ютер з встановленим пакетом AVR Studio.
2. Мікроконтролер моделі AT90S2313 або ATtiny2313.
3. Налагоджувальний стенд з програматором.

Основні теоретичні відомості.

В основі ядра AVR лежить Розширена RISC архітектура, що об'єднує розвинений набір команд і 32 регістри загального призначення. Усі 32 регістри безпосередньо підключені до арифметико-логічного пристрою (АЛУ), що дає доступ до будь-яких двох регістрів за один машинний цикл. Така архітектура забезпечує десятикратний вигаш в ефективності коду в порівнянні з традиційними CISC мікроконтролерами.

Мікроконтролер моделі AT90S2313 має наступні характеристики: 2кБ завантажувальної flash-пам'яті програм; 128 байт енергонезалежної перепрограмованої постійної пам'яті (EEPROM); 15 ліній вводу/виводу загального призначення, організованих в два порти на 8 і 7 ліній; 32 робочих регістра; програмовані таймери/лічильники з режимом збігу; зовнішні та внутрішні переривання; універсальний програмований послідовний порт; програмований сторожовий таймер з вбудованим генератором; послідовний порт SPI для завантаження програм; два обраних програмно режиму низького енергоспоживання.

Всі пристрої введення / виводу і периферійні пристрої мікроконтролера розташовуються в просторі введення / виводу (табл. 1). Різні осередки цього простору доступні через команди IN і OUT, пересилають дані між одним з 32-х регістрів загального призначення і простором введення/виводу. До регістрів \$00..\$1F можна здійснювати побітовий доступ командами SBI і CBI. Значення окремого біта цих регістрів можна перевірити командами SBIC і SBIS. Додаткову інформацію з цього питання можна знайти в описі системи команд.

При використанні спеціальних команд IN, OUT, SBIS і SBIC, повинні використовуватися адреси \$00..\$3F. При доступі до регістру вводу/виводу як до комірки ОЗУ, до його адресою необхідно додати \$20. У таблиці 1 адреси регістрів в пам'яті даних наведені в дужках.

Таблиця 1. - Основні регістри простору введення / виводу

Адреса	Позначення	Фірмова назва	Призначення
\$3F(\$5F)	SREG	Status REGISTER	Регістр стану
\$3D(\$5D)	SPL	Stack Pointer low byte	Показчик стека, молодший байт
\$3B(\$5B)	GIMSK	General Interrupt mask register	Загальний регістр маски переривань
\$3A(\$5A)	GIFR	General Interrupt Flag Register	Загальний регістр прапорів переривань
\$39(\$59)	TIMSK	Timer-counter Interrupt mask register	Регістр маски переривань від таймера-лічильника
\$38(\$58)	TIFR	Timer-counter Interrupt Flag Register	Регістр прапора переривання таймера-лічильника
\$35(\$55)	MCUCR	MCU general Control Register	загальний регістр управління мікроконтролером
\$33(\$53)	TCCR0	Timer-Counter 0 Control Register	Таймер-лічильник 0 (8 біт)
\$32(\$52)	TCNT0	Timer-Counter 0 (8-біт)	Регістр а управління таймером лічильником 1
\$2F(\$4F)	TCCR1A	Timer-Counter 1 Control Register A	Регістр В управління таймером лічильником 1
\$2E(\$4E)	TCCR1B	Timer-Counter 1 Control Register B	Таймер-лічильник 1 старший байт
\$2D(\$4D)	TCNT1H	Timer-Counter 1 High byte	Таймер-лічильник 1 молодший байт
\$2C(\$4C)	TCNT1L	Timer-Counter 1 Low byte	Регістр збігу Т\с 1 старший байт
\$2B(\$4B)	OCR1H	Output Compare Register 1 High byte	Регістр збігу Т \ с 1 молодший байт
\$2A(\$4A)	OCR1L	Output Compare Register 1 Low byte	Регістр захоплення Т\с 1 старший байт
\$25(\$45)	ICR1H	T/C 1 Input Capture Register High byte	Регістр захоплення Т \ с 1 молодший байт
\$24(\$44)	ICR1L	T/C 1 Input Capture Register Low byte	Регістр управління сторожовим таймером
\$21(\$41)	WDTCR	WatchDog Timer Control Register	Регістр адреси енергонезалежної пам'яті
\$1E(\$3E)	EEAR	EEPROM Address Register	Регістр даних енергонезалежної пам'яті
\$1D(\$3D)	EEDR	EEPROM Data Register	Регістр управління енергонезалежної пам'яті
\$1C(\$3C)	EECR	EEPROM Control Register	Реєстр даних порту В
\$18(\$38)	PORTB	Data Register, Port B	Реєстр напрямку даних порту В
\$17(\$37)	DDRB	Data Direction	Висновки порту В

		Register Port B	
\$16(\$36)	PINB	Input pins, Port B	Реєстр даних порту D
\$12(\$32)	PORTD	Data Register, Port D	Реєстр напрямку даних порту D
\$11(\$31)	DDRD	Data Direction Register Port D	Реєстр напрямку даних порту D
\$10(\$30)	PIND	Input pins, Port D	Висновки порту D
\$0C(\$2C)	UDR	UART Data Register	Реєстр даних послідовного порту
\$0B(\$2B)	USR	UART Status Register	Реєстр стану послідовного порту
\$0A(\$2A)	UCR	UART Control Register	Реєстр управління послідовного порту
\$09(\$29)	UBRR	UART Baud Rate Register	Регістр швидкості послідовного порту
\$08(\$28)	ACSR	Analog Comparator Control and Status Register	Регістр управління і стану аналогового компаратора

Регістр статусу - *SREG* (друга назва – реєстр прапорів або реєстр статусу) розташований за адресою \$3F (\$5F) простору вводу/виводу. Кожен біт цього реєстру використовується як індикатор (прапор) певного стану або результату. Значення бітів цього реєстру змінюються автоматично після кожної команди, крім того, їх можна прочитати і записати за допомогою команд IN, OUT, LD, ST. Позначення бітів наведені в таблиці 2.

Таблиця 2 - Позначення бітів реєстру стану SREG

Номер біта	7	6	5	4	3	2	1	0
Назва біта (прапора)	I	T	H	S	V	N	Z	C
Можливість читання – R і запису – W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Значення після включення або скидання	0	0	0	0	0	0	0	0

Біт 7-I. Загальна роздільна здатність переривань. Для вирішення переривань цей біт повинен бути встановлений. Управління окремими перериваннями проводиться реєстрами маски переривань - GIMSK і TIMSK. Якщо біт скинутий (0), незалежно від стану GIMSK/TIMSK всі переривання заборонені. Біт I скидається апаратно після входу в обробник переривання і відновлюється командою RETI при виході з нього. Для вирішення обробки вкладених переривань програміст повинен встановлювати цей біт на початку відповідного обробника переривань командою SEI.

Біт 6 – T. Зберігання копіюемого біта. Команди копіювання бітів BLD (Bit Load) і BST (Bit Store) використовують цей біт як джерело і приймач оброблюваного біта. Біт з реєстру загального призначення може

бути скопійований в Т командою BST, а сам біт Т може бути скопійований в Біт регістра загального призначення командою BLD.

Біт 5 - Н: Прапор половинного перенесення. Цей прапор відображає перенесення з молодшої половини байта при деяких арифметичних операціях.

Біт 4-с. біт знака, $S = N \text{ XOR } V$. біт S завжди дорівнює виключає АБО між прапорами N (негативний результат) і V (переповнення Доповнення до двох).

Біт 3 – V. Прапор переповнення доповнення до двох. Цей прапор підтримує арифметику з числами в додатковому коді. Більш докладно про це можна прочитати в описі системи команд.

Біт 2 – N. Прапор негативного результату. Цей прапор відображає негативний результат різних арифметичних і логічних операцій.

Біт 1 – Z. Прапор нульового результату. Цей прапор відображає нульовий результат різних арифметичних і логічних операцій.

Біт 0 – C. Прапор переносу. Цей прапор відображає перенесення або позику в арифметичних і логічних операціях.

Регістри для роботи із зовнішніми висновками. У розглянутій моделі контролера зовнішні висновки об'єднані в два порти: V-на 8 ліній і D-на 7 ліній.

Порт В 8-розрядний двонаправлений порт введення / виводу. Для його обслуговування відведено три регістра в просторі вводу/виводу: регістр даних PORTB (\$18), регістр напрямку передачі даних - DDRB (\$17) і прямий доступ до висновків порту PINB (\$16). Адреса PINB призначений тільки для читання, в той час як регістр даних і регістр напрямки даних – для читання і запису.

Всі висновки порту мають окремо підключаються підтягуючі резистори. Виходи порту В можуть поглинати струм до 20 mA і безпосередньо керувати світлодіодними індикаторами. Крім основного призначення виводів порту В можуть бути призначені альтернативні функції, які наведені в таблиці 3.

Таблиця 3 - Альтернативні функції виводів порту В

Висновок	Альтернативна функція
PB0	AIN0 – позитивний вхід аналогового компаратора
PB1	AIN1 – негативний вхід аналогового компаратора
PB3	OC1 – вихід збігу таймера-лічильника 1
PB5	MOSI – вхід даних для завантаження пам'яті
PB6	MISO – вихід даних для читання пам'яті
PB7	SCK – вхід тактових імпульсів послідовного обміну

При використанні альтернативних функцій висновків регістри DDRB і PORTB повинні бути встановлені відповідно до опису альтернативних функцій. PINB не є регістром, за цією адресою здійснюється доступ до фізичних значень кожного з висновків порту B. При читанні PORTB, читаються дані з реєстру, які до того були туди записані, а при читанні PINB читаються логічні значення, присутні на висновках порту.

Всі 8 біт порту B при використанні для введення/виводу рівноцінні і можуть бути налаштовані незалежно один від одного. Кожен біт регістру DDRB вибирає напрямок передачі даних для виведення порту з тим же номером. Якщо біт в регістрі DDRB встановлений (1) - даний висновок налаштований як вихід, а якщо скинутий (0) - висновок сконфігурований як вхід. Якщо встановлений біт в PORTB і висновок сконфігурований як вхід, включається вбудований підтягуючий резистор, який забезпечує гарантовану логічну одиницю на даному вході якщо відповідний йому висновок нікуди не підключений. Для відключення резистора, PORTB_n (де n = 7,6...0-номер виводу) повинен бути скинутий (0) або висновок повинен бути сконфігурований як вихід. Вищесказане ілюструє Таблиця 4.

Таблиця 4 - Вплив Ddbn і PORTB_n на висновки порту B

DDB _n	PORTB _n	Режим	Підтягуючий режим	Коментар
0	0	Вхід	Ні	третій стан (Hi-Z)
0	1	Вхід	Так	PB _n джерело струму 0,03...0,1 mA
1	0	Вихід	Ні	вихід встановлений в 0
1	1	Вихід	Ні	вихід встановлений в 1

Для порту D використовуються три регістри в просторі введення / виводу: регістр даних порту-PORTD (\$12), регістр напрямку даних-DDRD (\$11) і висновки порту D-PIND (\$10). Регістри даних і напрямки даних можуть читатися і записуватися, а регістр PIND – тільки для читання. Порт D – за структурою і принципом дії аналогічний порту Ст. Відмінності полягають тільки в наборі альтернативних функцій, які наведені у таблиці 5, а також в тому, що порт D має не 8, а 7 зовнішніх ліній, при цьому старший біт кожного регістра порту D не використовується. Якщо висновки порту використовуються для обслуговування альтернативних функцій, вони повинні бути налаштовані на введення / виведення відповідно до опису функції.

Таблиця 5 – Альтернативні функції порту D

Виведення порту	Альтернативна функція
PD0	RXD – вхід даних послідовно порту UART або USART
PD1	TXD – вихід даних послідовно порту UART або USART
PD2	INT0 – вхід зовнішнього переривання 0
PD3	INT1 – вхід зовнішнього переривання 1
PD4	T0 – зовнішній лічильний вхід таймера лічильника 0
PD5	T1 – зовнішній лічильний вхід таймера лічильника 1
PD6	ICP – вхід захоплення таймера лічильника 1

Інші моделі контролерів сімейства AVR можуть мати іншу кількість портів і зовнішніх виводів. Наприклад, модель Attiny4 має один порт на 4 лінії, а модель Atmega2560 – 11 портів на 86 ліній.

Порядок виконання роботи.

1. Розробити програму відповідно до свого варіанту завдання (Таблиця 6).
2. Запустити пакет AVR Studio і ввести розроблену програму.
3. Перевірити роботу програми в симуляторі і при необхідності виправити помилки.
4. Підключити налагоджувальний стенд до послідовного порту комп'ютера і записати розроблену програму в мікроконтроллер.
5. Перевірити працездатність програми на налагоджувальному стенді.

Зміст звіту.

1. Тема і мета роботи.
2. Блок-схема алгоритму.
3. Текст програми.
4. Висновок.
- 5.

Таблиця 6 - Завдання до лабораторної роботи

Номер варіанту	Завдання
1	Порт D-вхід, порт в-вихід. Вивести на порт в число в двійковому коді Рівне $B = D - 0b0010001$
2	Порт D-вхід, порт в-вихід. Вивести на порт в число в двійковому коді Рівне $B = d \text{ or } 0b00011001$
3	Порт D-вхід, порт в-вихід. Вивести на порт у Значення $0b00010101$ якщо $D = \$ 55$, інакше вивести значення $0b11011011$
4	Порт D-вхід, порт в-вихід. Вивести на порт В Значення $\$ 16$ якщо d з'єднати $\$ 05$, інакше вивести значення $\$ 0F$
5	Порт D-вхід, порт в-вихід. Вивести на порт у Значення $0b11110001$ якщо $d \geq \$ 55$, інакше вивести значення $0b11011011$
6	Встановити $PB0$ і скинути $PB1$ якщо $D = 0$, інакше встановити $PB1$ і скинути $PB0$

7	Встановити PB0 і скинути PB1 якщо PD1 = 1 and PD4 = 1 , інакше встановити PB1 і скинути PB0
8	Порт D-вхід, порт в-вихід. Вивести на порт в число в двійковому кодi Рівне $V = D + \$ 30$
9	Порт D-вхід, порт в-вихід. Вивести на порт в число в двійковому кодi Рівне $V = D \text{ and } \$ 40$
10	Порт D-вхід, порт в-вихід. Вивести на порт в число в двійковому кодi Рівне $V = D \text{ хог } \$ 22$
11	Порт D-вхід, порт в-вихід. Вивести на порт у Значення 0b00010001 якщо $D < \$ 55$, інакше вивести значення 0b11011011
12	Порт D-вхід, порт в-вихід. Вивести на порт у Значення 0b00010001 якщо $D < \$ 55$, вивести \$ 7C якщо $D = \$ 55$, інакше вивести значення 0b11011011
13	Встановити PB0 і скинути PB1 якщо $D < \$D1$, інакше встановити PB1 і скинути PB0
14	Встановити PB0 і скинути PB1 якщо PD1 = PD4, інакше встановити PB1 і скинути PB0
15	Порт D-вхід, порт в-вихід. Вивести на порт в число в двійковому кодi Рівне $V = D - 0b0010001$
16	Порт D-вхід, порт в-вихід. Вивести на порт в число в двійковому кодi Рівне $V = d \text{ or } 0b00011001$
17	Порт D-вхід, порт в-вихід. Вивести на порт у Значення 0b00010101 якщо $D = \$ 55$, інакше вивести значення 0b11011011
18	Порт D-вхід, порт в-вихід. Вивести на порт В Значення \$ 16 якщо d з'єднати \$ 05, інакше вивести значення \$ 0F
19	Порт D-вхід, порт в-вихід. Вивести на порт у Значення 0b11110001 якщо $d \geq \$ 55$, інакше вивести значення 0b11011011
20	Встановити PB0 і скинути PB1 якщо $D = 0$, інакше встановити PB1 і скинути PB0
21	Встановити PB0 і скинути PB1 якщо PD1 = 1 and PD4 = 1 , інакше встановити PB1 і скинути PB0
22	Порт D-вхід, порт в-вихід. Вивести на порт в число в двійковому кодi Рівне $V = D + \$ 30$
23	Порт D-вхід, порт в-вихід. Вивести на порт в число в двійковому кодi Рівне $V = D \text{ and } \$ 40$
24	Порт D-вхід, порт в-вихід. Вивести на порт в число в двійковому кодi Рівне $V = D \text{ хог } \$ 22$
25	Порт D-вхід, порт в-вихід. Вивести на порт у Значення 0b00010001 якщо $D < \$ 55$, інакше вивести значення 0b11011011
26	Порт D-вхід, порт в-вихід. Вивести на порт у Значення 0b00010001 якщо $D < \$ 55$, вивести \$ 7C якщо $D = \$ 55$, інакше вивести значення 0b11011011
27	Встановити PB0 і скинути PB1 якщо $D < \$D1$, інакше встановити PB1 і скинути PB0
28	Встановити PB0 і скинути PB1 якщо PD1 = PD4, інакше встановити PB1 і скинути PB0

ПРАКТИЧНА РОБОТА № 3 ВИВЧЕННЯ МОЖЛИВОСТЕЙ ВБУДОВАНИХ ТАЙМЕРІВ-ЛІЧИЛЬНИКІВ.

Мета роботи: вивчити структуру і режими роботи вбудованих таймерів-лічильників мікроконтролера AT90S2312. Отримати практичні навички використання таймерів-лічильників в прикладних програмах.

Прилади та обладнання:

1. Персональний комп'ютер з встановленим AVR Studio.
2. Мікроконтролер моделі AT90S2312 або Attyni2313.
3. Налагоджувальний стенд з програматором.

Основні теоретичні відомості.

У моделях AT90S2313 і Attyni2313 передбачені два таймера-лічильника: 8-розрядний таймер-лічильник 0 і 16-розрядний таймер-лічильник 1. Кожен з таймерів індивідуально підключається до одного з виходів 10 розрядного попереднього дільника частоти. Обидва таймера можуть використовуватися як таймери з внутрішнім джерелом імпульсів або лічильники імпульсів надходять ззовні.

В якості джерела рахункових імпульсів для таймерів можна вибрати сигнал з тактовою частотою процесора (СК), імпульси з виходів попереднього дільника частоти (СК/8, СК/64, СК/256 або СК/1024) або імпульси з відповідного зовнішнього виводу. Крім того, таймери можуть бути зупинені, заборною проходження імпульсів на них.

- Таймери-лічильники можна використовувати для наступних цілей:
- підрахунок зовнішніх імпульсів, наприклад, від імпульсного датчика переміщень;
- вимірювання інтервалів часу між зовнішніми подіями, в тому числі повністю апаратним способом;
- завдання витримки часу для різних цілей, в тому числі з генеруванням апаратного переривання після закінчення заданого часу;
- генерація послідовності імпульсів з регульованою частотою;
- генерація послідовності імпульсів з регульованою шпаруватістю (широтно-імпульсний модулятор).

Восьмирозрядний таймер/лічильник 0 в контролері AT90S2313 має мінімум функцій, зокрема не має режимів захоплення і збіги, а цей же лічильник в контролері Attyni2313 володіє повним набором перерахованих функцій. Для налаштування режиму роботи таймера-лічильника 0 використовується регістр TCCR0. Прапор переповнення таймера знаходиться в регістрі TIFR. Дозвіл і заборона переривань від таймера

управляється регістром TIMSK. Власне кількість підрахованих імпульсів знаходиться в регістрі TCNT0. Значення бітів регістру TCCR0 наведено в таблиці 1.

Таблиця 1- Регістр управління таймером / лічильником 0-TCCR0

Номер біта	7	6	5	4	3	2	1	0
Назва біта (прапора)	-	-	-	-	-	CS02	CS01	CS00
Можливість читання – R і запису – W	R	R	R	R	R	R/W	R/W	R/W
Значення після включення або скидання	0	0	0	0	0	0	0	0

Біти 7..3 – в AT90S231 ці біти не використовуються і завжди читаються як 0. Біти 2,1,0 – CS02, CS01, CS00 – вибір джерела рахункових імпульсів. Ці біти задають коефіцієнт ділення тактової частоти контролера або підключення зовнішнього входу (див. таблицю 2).

При роботі таймера / лічильника від зовнішнього сигналу, зовнішній сигнал синхронізується з тактовим генератором контролера. Для правильної обробки зовнішнього сигналу, мінімальний час між сусідніми імпульсами має перевищувати період тактової частоти процесора як мінімум в 2,5 рази.

Таблиця 2 – Вибір джерела рахункових імпульсів

CS02	CS01	CS00	Опис
0	0	0	Таймер/лічильник зупинений
0	0	1	СК
0	1	0	СК/8
0	1	1	СК/64
1	0	0	СК/256
1	0	1	СК/1024
1	1	0	Зовнішній висновок T0, наростаючий фронт
1	1	1	Зовнішній висновок T0, спадаючий фронт

Регістр таймера-лічильника TCNT0 містить кількість підрахованих імпульсів. Розмір цього регістра дозволяє вести рахунок від 0 до $2^8 - 1 = 255$. Він реалізований як наростаючий лічильник з можливістю читання і запису. При запису таймера / лічильника, якщо присутні тактові імпульси, таймер / лічильник продовжує рахунок в наступному за операцією запису тактовому циклі таймера.

16-розрядний таймер-лічильник 1 може отримувати імпульси тактової частоти - СК, імпульси з попереднього дільника (СК/8, СК/64,

СК/256 або СК/1024), імпульси із зовнішнього виводу або бути зупинений відповідними установками регістра TCCR1B (див. таблицю 6) аналогічно таймеру-лічильнику 0 з регістром TCCR0. Прапори стану таймера (переповнення, збіги і захвату) і керуючі сигнали знаходяться в регістрі TIFR. Дозвіл і заборона переривань від таймера 1 управляється регістром TIMSK. Вимоги до зовнішніх лічильних імпульсів такі ж, як і для таймера-лічильника 0.

Таймер-лічильник 1 підтримує функцію збіги, використовуючи пару регістрів збігу OCR1H і OCR1L в якості джерела для порівняння з вмістом самого лічильника – парою регістрів TCNT1H і TCNT1L. Можливо включення функцій апаратного скидання в нуль лічильника і переключення виходу OC1 при настанні події збігу.

Таймер-лічильник 1 можна використовувати як 8-, 9 - або 10-розрядний широтно-імпульсний модулятор. У цьому режимі лічильник і регістри OCR1H, OCR1L працюють як захищений від брязкіта незалежний широтно-імпульсний модулятор з відцентрованими імпульсами. Детально ця функція буде описана нижче.

Функція захоплення входу передбачає копіювання вмісту таймера-лічильника 1 в регістр захоплення ICR1H, ICR1L і управляється зовнішнім сигналом на вході захоплення - ICP. Робота режиму захоплення визначається керуючим регістром TCCR1A (див. таблицю 3). При роботі захоплення входу, може бути включена схема придушення шуму, при цьому сигнал захоплення виникає тільки в тому випадку, якщо подія, яка керує захопленням спостерігається протягом 4-х машинних циклів.

Таблиця 3- Регістр А управління таймером-лічильником 1-TCCR1A

Номер біта	7	6	5	4	3	2	1	0
Назва біта(прапора)	COM1A1	COM1A0	-	-	-	-	PWM11	PWM10
Можливість читання – R і записи – W	R/W	R/W	R	R	R	R	R/W	R/W
Значення після включення та скидання	0	0	0	0	0	0	0	0

Біти 7,6 – COM1A1, COM1A0 – задають режим виходу збіги, тобто відгук виведення PB3 (OC1) на подію збігу регістру порівняння OCR1H, OCR1L і таймера-лічильника 1 TCNT1H, TCNT1L. Оскільки це альтернативна функція порту, відповідний біт напряму в регістрі DDRB повинен встановлювати висновок PB3 на вихід. Конфігурація керуючих біт показана в таблиці 4.

Таблиця 4- Встановлення режиму збігу

COM1A1	COM1A0	Опис
0	0	Таймер-лічильник 1 відключений від виводу OC1
0	1	Перемикання виходу OC1
1	0	Скидання (0) виводу OC1
1	1	Установка (1) виводу OC1

В режимі ШІМ ці біти мають інші функції, які вказані в таблиці 5. При зміні бітів COM1A1 і COM1A0 переривання за збігом повинно бути заборонено, очищенням відповідного біта в регістрі TIMSK. Інакше, переривання може статися під час зміни бітів. Біти 5..2 в AT90S2313 зарезервовані і завжди читаються як 0.

Біти 1,0-PWM11, PWM10-біти установки режиму широтно-імпульсної модуляції (ШІМ). Ці біти встановлюють режим роботи таймера-лічильника 1 в якості ШІМ (див. таблицю 5). Детальніше цей режим буде розглянутий нижче.

Таблиця 5 – Установка режиму роботи ШІМ

PWM11	PWM10	Опис
0	0	Робота ШІМ заборонена
0	1	8-розрядний ШІМ
1	0	9-розрядний ШІМ
1	1	10-розрядний ШІМ

Таблиця 6-регістр В управління таймером-лічильником 1-TCCR1B

Номер біта	7	6	5	4	3	2	1	0
Назва біта (прапора)	ICNC1	ICES1	–	–	CTC1	CS12	CS11	CS10
Можливість читання – R і запису – W	R/W	R/W	R	R	R/W	R/W	R/W	R/W
Значення після включення або скидання	0	0	0	0	0	0	0	0

Бит7 – ICNC1 – пригнічувач вхідного шуму входу захоплення. Якщо цей біт скинутий (0), придушення вхідного шумів входу захоплення заборонено. При цьому захоплення спрацьовує по першому заданому (наростаючому або спадаючому) фронту сигналу на вході ICP. При установці біта ICNC1 обробляються чотири послідовні вибірки сигналу на виводі ICP. Для спрацьовування захоплення всі вибірки повинні відповідати рівню заданому бітом ICES1. Частота вибірок дорівнює тактовій частоті контролера.

Бит 6 – ICES1 – вибір фронту сигналу захоплення. Якщо біт ICES1 скинутий (0) вміст таймера-лічильника 1 переписується в регістр захоплення по спадаючому фронту сигналу на виводі ICP. Якщо біт встановлений - то по наростаючому фронту сигналу.

Біти 5,4 в АТ90S2313 не використовуються і завжди читаються як 0.

Біт 3 – СТС1 – очищення таймера лічильника 1 за збігом. Якщо біт встановлений (1), таймер-лічильник 1 скидається в \$ 0000 в такті наступному за подією збігу. Якщо біт скинутий, таймер-лічильник 1 продовжує рахувати поки не буде зупинений, скинутий, відбудеться його переповнення або зміна напрямку рахунку. У режимі ШІМ цей біт не працює.

Біти 2,1,0 – CS12, CS11, CS10 – вибір джерела рахункових імпульсів. Ці біти визначають джерело лічильних імпульсів для таймера-лічильника 1 відповідно до таблиці 7.

Таблиця 7 – Вибір джерела рахункових імпульсів

CS12	CS11	CS10	Опис
0	0	0	Таймер / лічильник 1 зупинений
0	0	1	СК
0	1	0	СК/8
0	1	1	СК/64
1	0	0	СК/256
1	0	1	СК/1024
1	1	0	Спадаючий фронт на виведенні T1
1	1	1	Наростаючий фронт на виведенні T1

Оскільки таймер-лічильник 1 має розмір 16 розрядів і дозволяє вести рахунок від 0 до $2^{16} - 1 = 65535$, то результат рахунку цього лічильника розміщується у двох восьми розрядних регістрах: TCNT1H – старша частина і TCNT1L – молодша частина. Ця пара регістрів, що містить поточне значення таймера-лічильника 1. Щоб читання і запис двох регістрів лічильника відбувалося синхронно, для роботи з нею використовується тимчасовий регістр (TEMP), який є посередником між процесором і регістром TCNT1H. При запису старшого байта в регістр TCNT1H, записувані дані поміщаються в регістр TEMP. Потім, при записі молодшого байта в TCNT1L, він разом з даними з TEMP переписується в таймер-лічильник 1. Таким чином, при запису 16-розрядного значення першим повинен записуватися старший байт в TCNT1H, а потім молодший – в TCNT1L. При читанні молодшого байта з TCNT1L, він надсилається на процесор, і одночасно дані з TCNT1H переписуються в регістр TEMP, тобто одночасно читаються всі 16 розрядів. При подальшому читанні регістра TCNT1H, дані в дійсності читаються з регістра TEMP, тобто, при читанні порядок звернення до регістрів лічильника повинен бути зворотним: спочатку читається молодший байт з TCNT1L, а потім старший – з TCNT1H.

Таймер-лічильник 1 організований як наростаючий лічильник (в режимі ШІМ - наростаючий/спадання) з можливістю читання і запису.

Якщо вибрано джерело лічильних імпульсів для таймера / лічильника 1, то після запису в нього нового значення, він продовжує рахунок в наступному після операції запису періоді лічильного сигналу.

Регістр збігу OCR1AH, OCR1AL – 16-розрядний регістр, доступний для читання і запису. В цьому реєстрі зберігаються дані, які безперервно порівнюються з поточним значенням таймера-лічильника 1. Дія за збігом задається регістрами управління таймером-лічильником 1 і регістром стану. Оскільки регістр OCR1A є 16-розрядним, при записі в нього нового значення послідовність доступу до його старшої і молодшої частин повинна бути аналогічна вище розглянутої послідовності для пари регістрів TCNT1H і TCNT1L.

Регістр захоплення ICR1H, ICR1L – 16-розрядний регістр доступний тільки для читання. По наростаючому або спадаючому фронту (у відповідності з вибором фронту імпульсу захоплення ICES1) сигналу на виводі ICP поточне значення таймера-лічильника 1 копіюється в регістр захоплення ICR1H, ICR1L. В цей же час встановлюється прапор захоплення ICF1 в регістрі TIFR. Оскільки регістр захоплення є 16-розрядним, для коректного читання його значення першим повинен читатися молодший байт ICR1L, а другим – старший ICR1H.

При виборі режиму широтно-імпульсної модуляції (ШІМ), таймер/лічильник 1 і регістр збігу OCR1A формують 8, 9 або 10-розрядний безперервний вільний від "тремтіння" і правильний по фазі сигнал, що виводиться на ніжку PB3(OC1). Таймер-лічильник 1 в цьому режимі працює як Реверсивний лічильник, що рахує від 0 до кінцевого значення (див. таблицю 8). При досягненні кінцевого значення лічильник починає вважати в зворотну сторону до нуля, після чого робочий цикл повторюється. Коли значення лічильника збігається з 8, 9 або 10-ма молодшими бітами регістру OCR1A, висновок PD1 (OC1) встановлюється або скидається відповідно до установками біт COM1A1 і COM1A0 в регістрі TCCR1A (див. таблицю 9).

Таблиця 8 – Кінцеве значення таймера і частота ШІМ.

Дозвіл ШІМ	Кінцеве значення таймера	Частота ШІМ
8 біт	\$00FF (255)	Ftc1/510
9 біт	\$01FF (511)	Ftc1/1022
10 біт	\$03FF (1023)	Ftc1/2046

В режимі ШІМ, при записі в регістр OCR1A, 10 молодших біт передаються у тимчасовий регістр і переписуються тільки при досягненні таймером-лічильником кінцевого значення. При цьому усувається поява несиметричних імпульсів (тремтіння), які неминучі при асинхронної запису OCR1A.

Таблиця 9 – Установка режиму збігу при роботі ШІМ

COM1A1	COM1A0	Вплив на виведення ОС1
0	0	не підключений
0	1	не підключений
1	0	очищається при збігу, для зростання лічильника і скидається для зменшення (неінвертуючий ШІМ)
1	1	очищається при збігу, для зменшення лічильника і скидається для зростання (інвертується ШІМ)

Якщо OCR1A містить значення \$0000 або кінцеве значення (див. таблицю 8), висновок ОС1 залишається в одному логічному стані, яка визначається установками COM1A1 і COM1A0. В режимі ШІМ прапор переповнення таймера 1 (TOV1) в регістрі TIFR встановлюється, коли лічильник змінює напрям рахунку в точці \$0000. Переривання по переповненню таймера 1 працює як при нормальному режимі роботи таймера-лічильника, тобто воно виконується, якщо встановлений прапор TOV1 і можна відповідні переривання. Те ж саме стосується прапора збігу і переривання за збігом.

Порядок виконання роботи.

1. Розробити керуючу програму відповідно до завдання (Таблиця 10).
2. Запустити пакет AVR Studio і ввести розроблену програму.
3. Перевірити роботу програми в симуляторі і при необхідності виправити помилки.
4. Підключити налагоджувальний стенд до послідовного порту комп'ютера і записати розроблену програму в мікроконтроллер.
5. Перевірити працездатність програми на налагоджувальному стенді.

Зміст звіту.

1. Тема і мета роботи.
2. Блок-схема алгоритму.
3. Текст програми.
4. Висновок.

Таблиця 10-Завдання до лабораторної роботи

Номер варіанту	Завдання
1	Генератор імпульсів регульованої частоти зі шпаруватістю 2. Імпульси виводити на висновок PD1, період імпульсів задається прямим двійковим кодом на порт в межах 0,065536 сек ... 16,777216 сек.
2	Генератор імпульсів регульованої частоти зі шпаруватістю 2.

	Імпульси виводити на висновок PD1, період імпульсів задається прямим двійковим кодом на порт в в межах 0,1 сек ... 25,5 сек з кроком 0,1 сек.
3	Генератор імпульсів регульованої частоти зі шпаруватістю 2. Імпульси виводити на висновок PD1, період імпульсів задається прямим двійковим кодом на порт в в межах 1 сек ... 255 сек з кроком 0,1 сек.
4	Пристрій витримки часу. Запуск інтервалу виробляти негативним фронтом на вході PD6. З моменту запуску на виведенні PD1 сформувати імпульс високого рівня, тривалість якого задається прямим двійковим кодом на порт в в межах 0,1 ... 25,5 с.
5	Пристрій витримки часу. Запуск інтервалу виробляти негативним фронтом на вході PD6. З моменту запуску на виведенні PD1 сформувати імпульс високого рівня, тривалість якого задається прямим двійковим кодом на порт в в межах 1 ... 255 с.
7	Пристрій вимірювання часу. Запуск вимірювання проводити негативним фронтом на вході PD6 і вимірювати час перебування на цьому вході низького логічного рівня. Результат вимірювань виводити в прямому двійковому кодої на порт в з точністю 0,1 с.
8	Пристрій вимірювання часу. Запуск вимірювання проводити негативним фронтом на вході PD6 і вимірювати час перебування на цьому вході низького логічного рівня. Результат вимірювань виводити в прямому двійковому кодої на порт в з точністю 1 с.
9	Пристрій вимірювання часу. Запуск вимірювання проводити негативним фронтом на вході PD6 і вимірювати час перебування на цьому вході низького логічного рівня. Результат вимірювань виводити в прямому двійковому кодої на порт в з точністю 0,5 с
10	Генератор імпульсів регульованої шпаруватості. Імпульси виводити на висновок PD1, період імпульсів 16,777216 с, шпаруватість задається прямим двійковим кодом на порт в в межах 2 ... 255 з кроком 1.
12	Генератор імпульсів регульованої шпаруватості. Імпульси виводити на висновок PD1, період імпульсів 2 с, шпаруватість задається прямим двійковим кодом на порт в в межах 2 ... 255 з кроком 1.
13	Генератор імпульсів регульованої шпаруватості. Імпульси виводити на висновок PD1, період імпульсів 256 с, шпаруватість задається прямим двійковим кодом на порт в в межах 2 ... 255 з кроком 1.

14	Широтно-імпульсний модулятор з роздільною здатністю 8 біт. Вихідний сигнал вивести на виведення PB3 (OC1). Частота ШІМ 7812,5 Гц. Значення шпаруватості задавати прямим двійковим кодом з порту D (7 біт) і лінії PB0 (8-й біт).
15	Широтно-імпульсний модулятор з роздільною здатністю 8 біт. Вихідний сигнал вивести на виведення PB3 (OC1). Частота ШІМ 30,51 Гц. Значення шпаруватості задавати прямим двійковим кодом з порту D (7 біт) і лінії PB0 (8-й біт).
16	Широтно-імпульсний модулятор з роздільною здатністю 8 біт. Вихідний сигнал вивести на виведення PB3 (OC1). Частота ШІМ 7,63 Гц. Значення шпаруватості задавати прямим двійковим кодом з порту D (7 біт) і лінії PB0 (8-й біт).
17	Лічильник зовнішніх подій. Як рахункового використовувати негативний імпульс на виведенні PD5 (T1). Результат рахунку виводити на порт в в прямому двійковому коді.
18	Лічильник зовнішніх подій зі скиданням. Як рахункового використовувати негативний імпульс на виведенні PD5 (T1). Результат рахунку виводити на порт в в прямому двійковому коді. Скидання лічильника за негативним імпульсом на виведенні PD2.
19	Лічильник зовнішніх подій. Як рахункового використовувати негативний імпульс на виведенні PD5 (T1). Результат рахунку виводити на порт в в унітарному коді коді.
20	Лічильник зовнішніх подій зі скиданням. Як рахункового використовувати негативний імпульс на виведенні PD5 (T1). Результат рахунку виводити на порт в в унітарному коді. Скидання лічильника за негативним імпульсом на виведенні PD2.
21	Перемикач "біжучі вогні" 1 з 8. Перемикаються виходи-порт В. період перемикання задається прямим двійковим кодом на порт D (7 біт) в межах 0,05 ... 6,35 с.
22	Перемикач "біжучі вогні" 1 з 8. Перемикаються виходи-порт В. Частота перемикання задається прямим двійковим кодом на порт D (7 біт) в межах 0,0787 ... 10 Гц
23	Перемикач "біжучі вогні" 2 з 8. Перемикаються виходи-порт В. період перемикання задається прямим двійковим кодом на порт D (7 біт) в межах 0,05 ... 6,35 с.
24	Перемикач "біжучі вогні" 1 з 3. Перемикаються виходи-порт В. період перемикання задається прямим двійковим кодом на порт D (7 біт) в межах 0,05 ... 6,35 с.
25	Перемикач "біжучі вогні" 2 з 3. Перемикаються виходи-порт В. період перемикання задається прямим двійковим кодом на порт D (7 біт) в межах 0,05 ... 6,35 с.
26	Пристрій управління чотирифазним кроковим двигуном.

	Виходи фаз: PD0, PD1, PD2, PD3. Режим роботи – 1-1. Період перемикання задається прямим двійковим кодом на порт в межах 0,05 ... 12,7 с.
27	Пристрій управління чотирифазним кроковим двигуном. Виходи фаз: PD0, PD1, PD2, PD3. Режим роботи – 2-2. Період перемикання задається прямим двійковим кодом на порт в межах 0,05 ... 12,7 с.
28	Пристрій управління чотирифазним кроковим двигуном. Виходи фаз: PD0, PD1, PD2, PD3. Режим роботи. Період перемикання задається прямим двійковим кодом на порт в межах 0,05 ... 12,7 с.
29	Пристрій управління чотирифазним кроковим двигуном із зовнішнім тактуванням. Виходи фаз: PD0, PD1, PD2, PD3. Режим роботи – 1-1. Тактовий вхід PD4.
30	Пристрій управління чотирифазним кроковим двигуном із зовнішнім тактуванням і реверсом. Виходи фаз: PD0, PD1, PD2, PD3. Режим роботи – 1-1. Тактовий вхід PD4. Вхід напрямку-PD6

ПРАКТИЧНА РОБОТА № 4 ВИВЧЕННЯ СИСТЕМИ ПЕРЕРИВАНЬ.

Мета роботи: вивчити структуру і режими роботи вбудованої системи переривань мікроконтролерів сімейства AVR. Отримати практичні навички використання переривань в прикладних програмах.

Прилади та обладнання:

1. Персональний комп'ютер з встановленим AVR Studio.
2. Мікроконтролер моделі AT90S2312 або ATtiny2313.
3. Налагоджувальний стенд з програматором.

Основні теоретичні відомості.

Перериванням називають призупинення виконання основної програми з деякого апаратного події і перехід на виконання спеціальної підпрограми обробки переривання, по завершенні якої управління повертається в основну програму, в то місце де вона була перервана. Адреса початку кожної підпрограми обробки переривання називають вектором переривання. Система переривань влаштована так, що при виникненні певного апаратного події поточне значення лічильника команд зберігається в стеку і відбувається безумовний перехід на один з перших адрес пам'яті програм – вектор переривання. Якщо програмісту потрібно використовувати дане переривання, то в цій комірці повинна бути поміщена команда безумовного переходу на початок відповідної підпрограми обробки переривання, яка і є вектором переривання.

У AT90S2313 передбачено 10 джерел переривань (таблиця 1), в ATtiny2313 – 19 (таблиця 2). Цим перериванням відповідають певні вектори в початковій області пам'яті програм, яка називається таблицею векторів переривань. Кожному з переривань також присвоєно окремий біт в регістрах масок переривань GIMSK і TIMSK, що дозволяє дане переривання. Переривання виконується тільки якщо встановлений відповідний йому біт в регістрі маски переривань і встановлений біт I в регістрі стану SREG, який дозволяє загальне обслуговування переривань. Крім того, кожному переривання відповідає окремий біт в регістрах прапорів переривань GIFR (в ATtiny2313 EIFR) і TIFR, який автоматично встановлюється при виникненні відповідної події і автоматично скидається при вході у відповідну підпрограму обробки переривання.

Порядок розташування переривань в таблиці векторів визначає пріоритет різних переривань. Якщо одночасно виникає дві і більше події, що викликають переривання, то першим обробляється переривання з більш високим пріоритетом. Менші адреси відповідають більш високому рівню пріоритету. Найвищий рівень у скидання, наступний пріоритет у INT0 - зовнішнього запиту переривання 0 і т. д. В контролерах AT90S2313,

АТtyni2313 і більшості подібних пріоритети переривань визначені жорстко і не можуть бути змінені.

Таблиця 1. – Джерела переривань в контролері АТ90S2313

Номер вектора	Джерело переривання
0	Включення живлення або скиданн
1	Зовнішній запит переривання IRQ0
2	Зовнішній запит переривання IRQ1
3	Захоплення таймера-лічильника 1 по сигналу на висновку ICP
4	Подія збігу таймера-лічильника 1
5	Переповнення таймера-лічильника 1
6	Переповнення таймера-лічильника 0
7	Прийнятий байт через послідовний інтерфейс UART
8	Регістр даних UDR послідовного інтерфейсу UART звільнився
9	Передача байта через послідовний інтерфейс UART завершена
10	Змінився стан аналогового компаратора

Таблиця 2. – Джерела переривань в контролері АТtyni2313

Номер вектора	Джерело переривання
0	Включення живлення або скидання
1	Зовнішній запит переривання IRQ0
2	Зовнішній запит переривання IRQ1
3	Захоплення таймера-лічильника 1 по сигналу на висновку ICP
4	Подія збігу таймера-лічильника 1 в каналі А
5	Переповнення таймера-лічильника 1
6	Переповнення таймера-лічильника 0
7	Прийнятий байт через послідовний інтерфейс USART
8	Регістр даних UDR послідовного інтерфейсу USART звільнився
9	Передача байта через послідовний інтерфейс USART завершена
10	Змінився стан аналогового компаратора
11	Змінився стан зовнішніх висновків
12	Подія збігу таймера-лічильника 1 в каналі В
13	Подія збігу таймера-лічильника 0 в каналі А
14	Подія збігу таймера-лічильника 0 в каналі В
15	Виявлено стартову умову для універсального послідовного інтерфейсу USI
16	Переповнення при прийомі чергового слова через інтерфейс USI
17	Енергонезалежна пам'ять EEPROM готова до запису чергового байта
18	Переповнення сторожового таймера

Для того, щоб використовувати механізм переривань, програміст в розроблюваній програмі повинен виконати наступні підготовчі дії:

1. оформити таблицю векторів переривань;
2. розробити підпрограми обробки потрібних йому переривань;

3. встановити в регістрах масок переривань прапори, що дозволяють обробку потрібних переривань;

4. після завершення всіх налаштувань мікроконтролера встановити прапор I в регістрі стану SREG

Нижче наведено приклад оформлення початку програми з таблицею векторів переривань для контролера моделі AT90S2313:

```
$000 rjmp RESET ; Обробка скидання  
$ 001 rjmp EXT_INT0; обробка IRQ0  
$ 002 rjmp EXT_INT1; обробка IRQ1  
$003 rjmp TIM_CAPT1 ; Обробка захоплення таймера 1  
$004 rjmp TIM_COMP1; Обробка збігу таймера 1  
$005 rjmp TIM_OVF1 ; Обробка переповнення таймера 1  
$006 rjmp TIM_OVF0 ; Обробка переповнення таймера 0  
$007 rjmp UART_RXC ; Обробка прийому байта  
$008 rjmp UART_DRE ; Обробка звільнення UDR  
$009 rjmp UART_TXC ; Обробка передачі байта  
$00a rjmp ANA_COMP ; Обробка аналогового компаратора  
$00b MAIN: ldi r16,0x0F ; Початок основної програми
```

Лівий стовпець з адресами може бути опущений, а імена міток переходів змінені на більш зручні для розробника.

Якщо з усього набору використовуються не всі переривання (найбільш частий випадок), то для не використовуваних переривань у відповідних їм вектори повинні бути розміщені команди повернення з переривання RETI.

Під час скидання всі регістри введення/виводу встановлюються початкові значення, програма починає виконуватися з адреси \$000, за цією адресою повинна бути записана команда RJMP – відносний перехід на початок основної програми. Якщо в програмі не дозволяються переривання і вектори переривань не використовуються, у перших адресах пам'яті може бути записана основна програма. Порядок розміщення основної програми, підпрограм і підпрограм обробки преривний принципі може бути довільним. Необхідно тільки мати на увазі, що максимальна дальність переходу по команді RJMP становить +/- 64команди. Тому якщо розмір коду від таблиці векторів переривань до початку основної програми або якої-небудь підпрограми перевищує цю величину, то будуть потрібні додаткові мітки з командами RJMP, або використання команди JMP, яка доступна не у всіх моделях мікроконтролерів (для AT90S2313, Attyni2313 недоступна).

Коли виникає переривання біт глобального дозволу переривань I скидається (нуль) і всі переривання забороняються. Підпрограма обробки переривання може встановити цей біт для вирішення вкладених

переривань, тобто обробки другого переривання до закінчення обробки першого переривання. Прапор дозволу переривань I встановлюється в 1 при виконанні команди виходу з переривання – RETI.

Для переривань включаються статичними подіями (напр. збіг значення лічильника/таймера 1 з регістром збігу) прапор переривання зводиться при виникненні події. Якщо прапор переривання очищений і присутній умова виникнення переривання, прапор не буде встановлений, поки не відбудеться наступна подія.

Коли програмний лічильник встановлюється на поточний вектор переривання для його обробки, відповідний прапор, згенерований перериванням, апаратно скидається. Деякі прапори переривання можуть бути скинуті записом логічної одиниці в біт відповідний прапору.

Мінімальний час реакції на будь-який з передбачених у процесорі переривань становить 4 періоди тактової частоти. Після чотирьох циклів викликається програмний вектор обробляє дане переривання. За ці 4 циклу програмний лічильник записується в стек, покажчик стека зменшується на 2. Програмний вектор являє собою відносний перехід на підпрограму обслуговування переривання і цей перехід займає 2 періоду тактової частоти. Якщо переривання відбувається під час виконання команди триває кілька циклів, то перед викликом переривання завершується виконання цієї команди.

Вихід з підпрограми обслуговування переривання також займає 4 періоди тактової частоти. За ці 4 періоди з стека відновлюється програмний лічильник. Після виходу з переривання процесор завжди виконує ще одну команду перерваної програми перш ніж обслужити будь відкладене переривання.

Зауважимо, що регістр стану SREG апаратно не обробляється процесором, як при виклику підпрограм, так і при обслуговуванні переривань. Якщо програма вимагає збереження SREG, воно повинно проводитися програмою користувача.

AT90S2313 і Attyni2313 мають два регістри маски переривань GIMSK-загальний регістр маски переривань і TIMSK-регістр маски переривань від таймерів-лічильників. У регістрі gimsk контролера AT90S2313 використовується тільки два старших біта, а в Attyni2313-три.

Біт 7 - INT1: Запит зовнішнього переривання 1. Коли цей біт встановлений, а також встановлений біт I регістру стану, дозволяється переривання від зовнішнього виводу PD3 (INT1). Біти управління режимом запуску переривання (ISC11 і ISC10) знаходяться в регістрі управління мікроконтролером (MCUCR) і визначають, за якою подією відпрацьовується переривання: по спадаючому або наростаючому фронту

або ж за рівнем. Активність на виведенні призводить до виникнення переривань, навіть якщо висновок налаштований як вихід. Це дозволяє реалізувати програмні переривання.

Біт 6 - INT0: Запит зовнішнього переривання 0. Коли цей біт встановлений, а також встановлений біт I регістру стану, дозволяється переривання від зовнішнього виводу PD2 (INT0). Біти управління режимом запуску переривання (ISC01 і ISC00) в регістрі управління мікроконтролером (MCUCR) визначають, за якою подією відпрацьовується переривання - по спадаючому або наростаючому фронту або ж за рівнем. Активність на виведенні призводить до виникнення переривань, навіть якщо висновок налаштований як вихід.

Біти 5..0 – в AT90S2313 зарезервовані і завжди читаються як 0.

Біт 5 в Attyni2313 дозволяє переривання по зміні стану зовнішніх висновків. Дана функція в цій роботі не розглядається. При необхідності її опис можна знайти в оригінальній документації на мікроконтролер Attyni2313.

Загальний регістр прапорів переривань GIFR в AT90S2313 доступний для читання і запису і аналогічний за структурою регістру GIMSK, тобто в ньому використовуються тільки два старших біти, які слугують індикаторами настання відповідних подій.

Біт 7-INTF1-прапор зовнішнього переривання 1. При виникненні на виведенні INT1 події викликає переривання, INTF1 встановлюється в "1". Якщо встановлено біт I регістру SREG і Біт INT1 в GIMSK, відбувається перехід на вектор переривання за адресою \$ 002. Прапор очищається після виконання обробника переривання. Крім того, прапор можна очистити, записавши в нього логічну одиницю.

Біт 6-INTF0-прапор зовнішнього переривання 0. При виникненні на виведенні INT0 події викликає переривання, INTF0 встановлюється в "1". Якщо встановлено біт I регістру SREG і Біт INT0 в GIMSK, відбувається перехід на вектор переривання за адресою \$ 001. Прапор очищається після виконання обробника переривання. Крім того, прапор можна очистити, записавши в нього логічну одиницю.

Біти 5..0 – зарезервовані. і в AT90S2313 завжди читаються як 0.

У контролері моделі Attyni2313 аналогічний регістр називається EIFR і доповнений ще одним бітом. Біт 5 – PCIF – прапор переривання по зміні стану зовнішніх виводів порту Ст. Цей біт відображає зміну логічного рівня на будь-якому з висновків порту та ініціює відповідне переривання аналогічно вище розглянутим бітам

У регістрі маски переривань від таймерів-лічильників TIMSK використовується чотири біта, які дозволяють переривання по різних подіях від таймерів-лічильників

Біт 7 – TOIE1 – дозвіл переривання по переповненню таймера-лічильника 1. Якщо встановлено цей біт і біт дозволу переривань I в регістрі стану, то можна переривання по переповненню таймера-лічильника 1. Відповідне переривання (вектор \$005) виконується при переповненні таймера-лічильника 1. В регістрі прапорів таймерів-лічильників TIFR встановлюється прапор переповнення. Якщо таймер-лічильник 1 працює в режимі ШІМ, прапор переповнення встановлюється при зміні напрямку-рахунку, при значенні \$ 0000.

Біт 6-OCIE1A-Дозвіл переривання за збігом таймера-лічильника 1. Якщо встановлено цей біт і біт дозволу переривання в регістрі стану, то можна переривання за збігом таймера/лічильника 1. Переривання (вектор \$004) виконується при рівності таймера/лічильника 1 і регістра збігу. У регістрі TIFR при цьому встановлюється прапор збігу.

Біти 5,4 – зарезервовані і в AT90S2313 завжди читаються як 0.

Біт 3 – TICIE1 – Дозвіл переривання по входу захоплення. Якщо встановлений біт TICIE1 і біт дозволу переривання в регістрі стану, то можна переривання по входу захоплення. Відповідне переривання (вектор \$003) виконується за сигналом захоплення на виводі 11 (PD6/ICP). У регістрі TIFR при цьому встановлюється прапор захоплення.

Біт 2 – зарезервований і в AT90S2313 завжди читається як 0.

Біт 1 – TOIE0 – Дозвіл переривання по переповненню таймера-лічильника 0. Якщо цей біт встановлений в 1, біт I в регістрі стану встановлений в 1, то можна переривання по переповненню таймера/лічильника 0. При виникненні переповнення виконується відповідний вектор переривання (\$006) і прапор переповнення (TOV0) в регістрі прапорів переривань таймерів-лічильників (TIFR) встановлюється в 1.

Біт 0 – зарезервований і в AT90S2313 завжди читається як 0.

Регістр прапорів переривань від таймерів/лічильників – TIFR за структурою аналогічний регістру TIMSK, але використовувані в ньому біти 1, 3, 6 і 7 є індикаторами настання відповідних подій

Біт 7 – TOV1 – прапор переповнення таймера-лічильника 1. Прапор TOV1 встановлюється ("1") при виникненні переповнення таймера-лічильника і скидається апаратно при виконанні відповідного вектора обробки переривання. Крім того, прапор можна скинути, записавши в нього логічну одиницю. Якщо встановлений біт I в SREG і Біт TOIE1 в

TIMSK, то при установці біта TOV1 виконується переривання по переповненню таймера-лічильника 1. У режимі ШІМ цей біт встановлюється, коли таймер / лічильник 1 змінює напрямок рахунку при значенні \$ 0000.

Біт 6 – OCF1A – прапор виходу збігу 1A. Цей прапор встановлюється в "1" якщо відбувається збіг значення таймера-лічильника 1 і даних в регістрі OCR1A. Прапор очищається апаратно при виконанні відповідного вектора переривання. Крім того, прапор можна скинути, записавши в нього логічну одиницю. Якщо встановлений біт I в SREG і Біт OCIE1A в TIMSK, то при установці біта OCF1A виконується переривання.

Біти 5,4 – зарезервовані в AT90S2313 завжди читаються як 0.

Біт 3 – ICF1 – прапор входу захоплення 1. Цей біт встановлюється в "1" при виникненні події захоплення по входу. Він відображає, що значення таймера-лічильника 1 скопійовано в регістр захоплення по сигналу на вході ICR1. ICF1 очищається апаратно при виконанні відповідного вектора обробки переривання. Крім того, прапор можна очистити, записавши в нього логічну одиницю.

Біт 2 – зарезервований і в AT90S2313 завжди читається як 0.

Біт 1 – TOV0 – прапор переповнення таймера лічильника 1. Прапор TOV0 встановлюється в "1" при переповненні таймера-лічильника 0 і скидається апаратно при виконанні відповідного вектора переривання. Крім того, прапор можна очистити, записавши в нього логічну одиницю. Якщо встановлений біт I в SREG і Біт TOIE0 в TIMSK, то при установці біта TOV0 виконується переривання по переповненню таймера-лічильника 0.

Біт 0 – зарезервований і в AT90S2313 завжди читається як 0.

Регістр управління мікроконтролером-MCUCR_p містить біти загального управління мікроконтролером.

Біти 7,6 зарезервовані і в AT90S2313 завжди читаються як 0.

Біт 5 – SE – дозвіл режиму Sleep. Цей біт повинен бути встановлений в 1, щоб при виконанні команди SLEEP процесор переходив в режим зниженого енергоспоживання. Цей біт повинен бути встановлений в 1 до виконання команди SLEEP.

Біт 4 – SM – Режим Sleep. Цей біт вибирає один з двох режимів зниженого енергоспоживання. Якщо біт скинутий (0), то в якості режиму Sleep вибирається холостий режим (Idle mode). Якщо біт встановлений, – вибирається економічний режим (Power down). Особливості кожного з режимів можна знайти в оригінальній документації на мікроконтролери.

Біти 3,2-ISC11, ISC10-біти управління спрацьовуванням переривання 1. Зовнішнє переривання активується виводом INT1 якщо встановлений прапор I регістру стану SREG і встановлена відповідна маска в регістрі GIMSK. Вплив бітів ISC11, ISC10 на спрацьовування переривання по рівню або фронтах наведено в таблиці 3

Таблиця 3 – Управління спрацьовуванням переривання INT1.

ISC11	ISC10	Опис
0	0	Запит на переривання по низькому рівню на вході INT1
0	1	Зарезервованій
1	0	Запит на переривання по спадаючому фронту на вході INT1
1	1	Запит на переривання по наростаючому фронту на вході INT1

При зміні бітів ISC11/ISC10 переривання INT1 повинно бути заборонено очищенням відповідного біта в регістрі GIMSK. Інакше переривання може виникнути під час зміни бітів.

Біти 1,0-ISC01, ISC00-біти управління спрацьовуванням переривання 0. Зовнішнє переривання активується висновком INT0, якщо встановлений прапор I регістру стану SREG і встановлена відповідна маска в регістрі GIMSK. У таблиці 4 наведено встановлення бітів для завдання спрацьовування за рівнем або фронтами.

Таблиця 4 – Управління спрацьовуванням переривання INT0.

ISC01	ISC00	Опис
0	0	Запит на переривання по низькому рівню на вході INT0
0	1	Зарезервованій
1	0	Запит на переривання по спадаючому фронту на вході INT0
1	1	Запит на переривання по наростаючому фронту на вході INT0

При зміні бітів ISC01 і ISC00, переривання по входу INT0 повинні бути заборонені скиданням біта дозволу переривання в регістрі GIMSK. Інакше переривання може статися при зміні значення бітів.

Порядок виконання роботи.

1. Розробити керуючу програму відповідно до завдання.
2. Запустити пакет AVR Studio і ввести розроблену програму.
3. Перевірити роботу програми в симуляторі і при необхідності виправити помилки.
4. Підключити налагоджувальний стенд до послідовного порту комп'ютера і записати розроблену програму в мікроконтроллер.
5. Перевірити працездатність програми на налагоджувальному стенді.

Зміст звіту.

1. Тема і мета роботи.
2. Блок-схема алгоритму.
3. Текст програми.
4. Висновок.

Завдання до лабораторної роботи

В якості завдань до роботи використовувати завдання до практичної роботи №3, але поставлені завдання вирішити з використанням переривань.

ЛІТЕРАТУРА

1. Токхайм Р. Мікропроцесори.: Курс і вправи. Під ред. В. Н. Грасевича-М.: Енергомашвидав 1988-336с.
2. Євстіфєєв А. В. Мікроконтролери AVR сімейства Tiny і Mega фірми "ATMEL" .-2004 .-558 .-63.30
3. Бродін В. Б., Шагурін М. І. Мікроконтролери. Архітектура, програмування, інтерфейс:Довідник .-М.: ЕКОМ, 1999 .-400
4. Виглеб Г. Датчики: пристрій і застосування / Переклад з ньому. Арсен Мірзоян-Зимовий Пляж-М.: Світ, 1989 .-196
5. Поліванов В. А., Волкова Н. А. електронні та електро-механічні датчики Balluff для систем автоматизації / / світ техніки та технологій.-2005.-9 .- 48-51
6. Москаленко В. В. електричний привід: Навчальний посібник .- М.:Майстерність, 2001 .-368
7. Гумен В. Ф., Калінінська Т. В. Стежить кроковий електродвигун .- Л.: Енергія,1980 .-168
8. Єгоров В. Ф. , Єгоров С. В. Електропривод металургійних машин і агрегатів:Навчальний посібник .-Новокузнецьк,2002 .-336
9. AT90S2313 Datasheet. Atmel
10. Сосонкін В. П. Мікропроцесорні системи ЧПУ. М.:Машинобудування 1985.-288с.
11. Ратмиров Ст. А. Управління верстатами гнучких виробничих систем – М:Машинобудування, 1986.