

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Тернопільський національний технічний університет**  
**імені Івана Пулюя**

**Кафедра автоматизації технологічних  
процесів та виробництв**

**Методичні вказівки**  
**до лабораторної роботи №4**  
**«Проектування електричної схеми в середовищі Proteus»**  
**з дисципліни**  
**«Проектування мікропроцесорних систем керування**  
**технологічними процесами»**

**Тернопіль 2022**

Методичні вказівки до лабораторної роботи №4 "Проектування електричної схеми в середовищі Proteus" з курсу "Проектування мікропроцесорних систем керування технологічними процесами"/ Медвідь В.Р., Пісьціо В.П. – Тернопіль: ТНТУ, 2022. – 14 с.

Відповідальні за випуск

доцент, к.т.н. Медвідь В.Р.,  
асистент Пісьціо В.П.

Методичні вказівки призначені для студентів спеціальності 151 "Автоматизація та комп'ютерно-інтегровані технології", котрі вивчають курс "Проектування мікропроцесорних систем керування технологічними процесами". Методичні вказівки можуть бути корисні для інших приладобудівних спеціальностей, а також для інженерів, що займаються розробкою мікропроцесорних систем на основі Arduino.

#### Лабораторна робота № 4

Проектування систем керування в середовищі Proteus VSM. Керування кроковими двигунами

##### Використання контролерів L297 та L298N для керування кроковими двигунами

Мікросхема L297 призначена для використання з подвійним мостовим драйвером (L298N), квадратною матрицею Дарлінгтона або дискретними пристроями в системах управління кроковими електродвигунами. Вона отримує сигнали синхронізації кроками двигуна, напрямки та режиму роботи від зовнішнього мікроконтролера та генерує сигнали управління для силових каскадів.

Інтегральний контролер крокового двигуна L297 формує чотири фазові сигнали для двофазних біполярних і чотирифазних уніполярних крокових електродвигунів у системах, керованих мікроконтролерами.

Із контролером двигун може працювати в режимах: повнокроковому, напівкроковому, коливальному.

Вбудований в контролер L297 переривач широтно-імпульсної модуляції (ШІМ) дозволяє здійснювати імпульсне керування струмом в обмотках крокового двигуна. Особливістю даного пристрою є те, що йому потрібні, тільки вхідні сигнали синхронізації, напрямку обертання та режиму роботи.

Оскільки фази генеруються всередині контролера, програмне навантаження мікроконтролера значно знижується.

##### Призначення виводів L297

Номер	Позначення	Призначення
1	SYNC	Вхід/вихід. Вивід синхронізації з іншими м/с L297.
2	GND	Загальний вивід
3	HOME	Показує стан виходів ABCD=0101. Вихід типу "відкритий колектор".
4	A	Фаза "А". Сигнал керування силовим каскадом.
5	$\overline{\text{INH1}}$	Вихідний сигнал заборони роботи фаз "А" та "В". Активний низький рівень забороняє роботу драйвера цих фаз (L298).
6	B	Фаза "В". Сигнал керування силовим каскадом.
7	C	Фаза "С". Сигнал керування силовим каскадом.
8	$\overline{\text{INH2}}$	Вихідний сигнал заборони роботи фаз "С" та "D". Активний низький рівень забороняє роботу драйвера цих фаз (L298).
9	D	Фаза "D". Сигнал керування силовим каскадом.
10	ENABLE	Вхід роздільної здатності.
11	CONTROL	Вхід керування режимом роботи ШІМ переривника. При низькому рівні переривника діють виходи INH1 і INH2. При високому рівні переривника діють фазові виходи А, В, С, D.
12	Vs	Живлення +5V
13	SENS2	Вхідний сигнал контролю струму навантаження силового каскаду фаз "С" та "D".
14	SENS1	Вхідний сигнал контролю струму навантаження силового каскаду фаз "А" та "В".
15	Vref	Опорна напруга для ШІМ переривника. Напруга, прикладена до цього виводу, визначає піковий струм навантаження.
16	OSC	Вхід підключення частото задаючого RC-ланцюга ШІМ переривача. $f \sim 1/0.69 RC$ .
17	CW/CCW	Вхід, що задає напрямок обертання двигуна.
18	CLOCK	Вхідний сигнал, що тактує кроки двигуна.
19	HALF/ $\overline{\text{FULL}}$	Вхід задає режими половинного (лог.1) чи повного (лог.0) кроку.
20	$\overline{\text{RESET}}$	Вхід скидання. Активний низький рівень переводить мікросхему у стан виходів ABCD=0101

## Гранично допустимі режими L297

Максимальна напруга живлення $V_s$	10V
Максимальна вхідна напруга	7V
Максимальна розсіювана потужність	1W
Діапазон температур кристала	-40..+150°C

## Основні параметри L297:

Параметр	MIN	TYP	MAX
Напруга живлення $V_s$	4,75V	5V	7V
Вхідна напруга низького рівня (L)			0,6V
Вхідна напруга високого рівня (H)	2,0V		$V_{ss}$
Струм споживання $I_s$		50mA	80mA
Вхідний струм (L)		100μA	
Вхідний струм (H)			10μA
Вихідна напруга низького рівня (A,B,C,D,INH)			0,4V
Вихідна напруга високого рівня (A, B, C, D, INH)		3,9V	
Вхідна опорна напруга $V_{ref}$		0	3V
Тактовий сигнал		0,5μS	
Час перемикання (кроку)		1μS	

Типова схема управління кроковим двигуном за допомогою комплексу мікросхем L297 та L298N показана на рис. Мікросхема L298N, по суті, є підсилювачем потужності, до виходів якої безпосередньо під'єднуються обмотки двигуна (рис. 1). Напруга живлення вихідних каскадів мікросхеми є більшою за напругу живлення контролера L297 (+5 В) і може досягати значень від +5 В до +46 В.

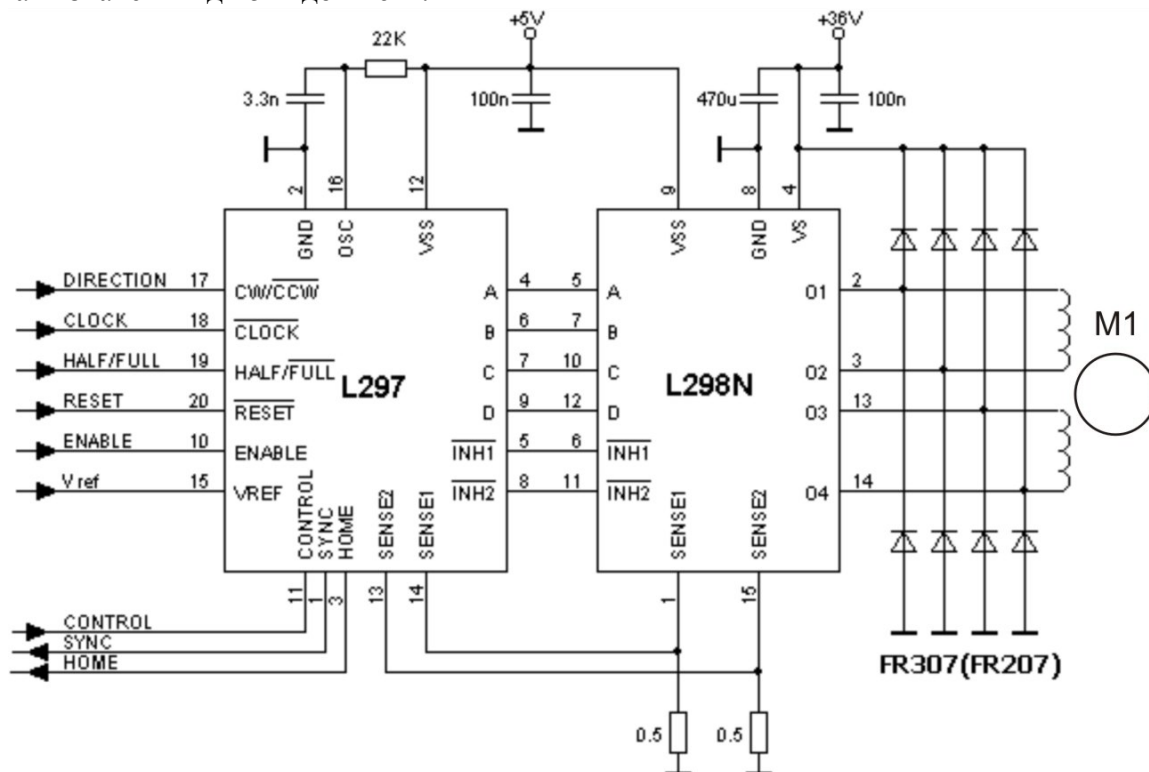


Рис. 1 Типова схема управління кроковим двигуном за допомогою мікросхем L297 та L298N

### Послідовність фаз руху двигуна

Контролер L297 генерує фазові групи для режимів повнокрокового, хвильового приводу та півкрокового режиму. Послідовність станів і вихідні сигнали для цих трьох режимів, показані нижче. У всіх випадках стан виводів контролера змінюється при переході від низького до високого рівня на вході CLOCK. Показано обертання за годинниковою стрілкою; для обертання проти годинникової стрілки послідовності просто змінюються на відповідь.



RESET встановлює контролер у стані, у якому ABCD = 0101.

### Півкроковий режим роботи двигуна

Режим половинного кроку вибирається високим рівнем на вході HALF/FULL. Часові діаграми сигналів на виході мікросхеми та зміна стану виходів A,B,C,D на кожному з півкроку показані на рис. 2.

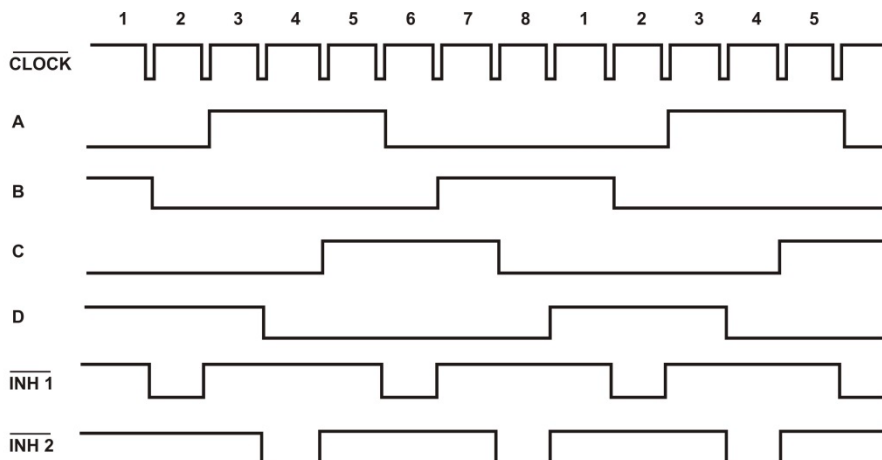
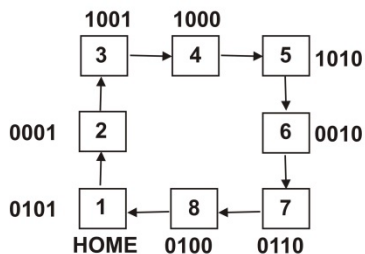


Рис. 2

### Повнокроковий режим роботи двигуна

Режим роботи двигуна (він ще має назву «двофазний») вибирається низьким рівнем на вході HALF/FULL, коли контролер знаходиться в непарному стані (1, 3, 5 або 7). У цьому режимі виходу INH1 і INH2 залишаються високими впродовж усього робочого циклу (рис. 3).

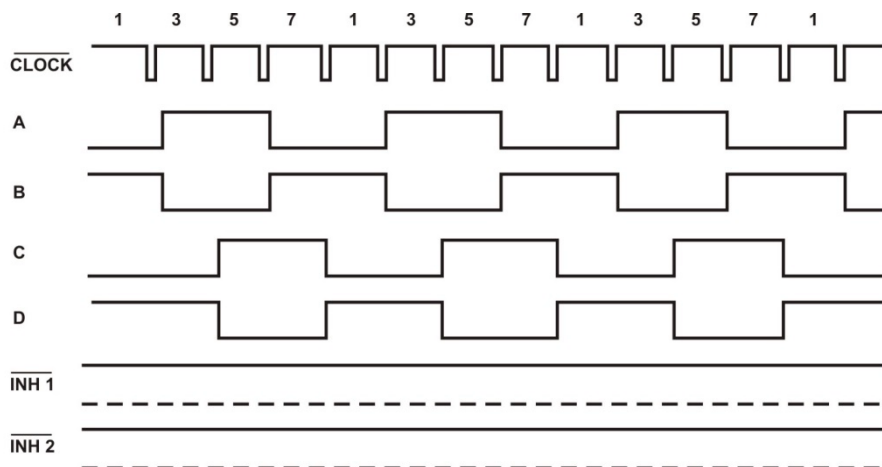
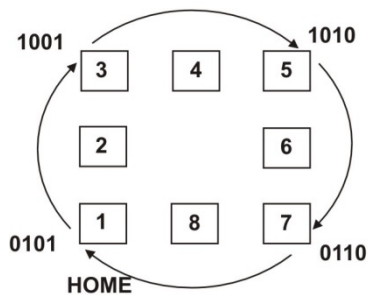


Рис. 3

### Коливальний режим роботи

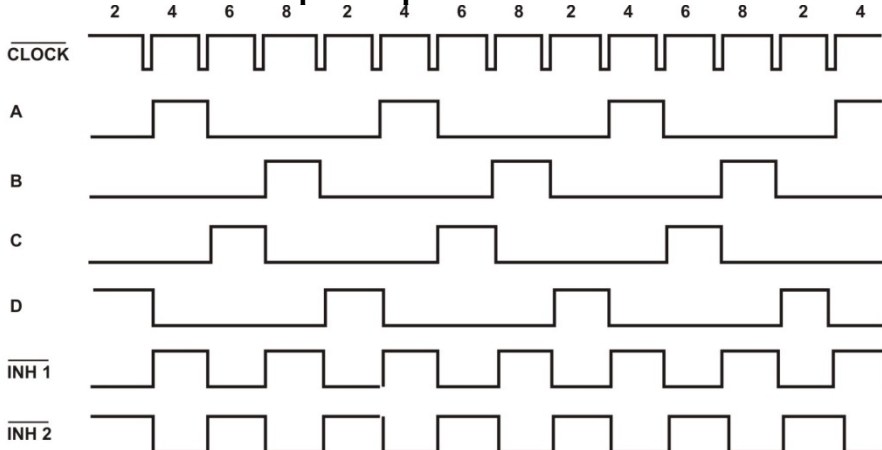
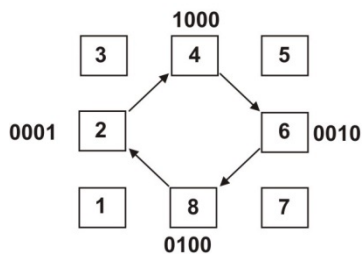



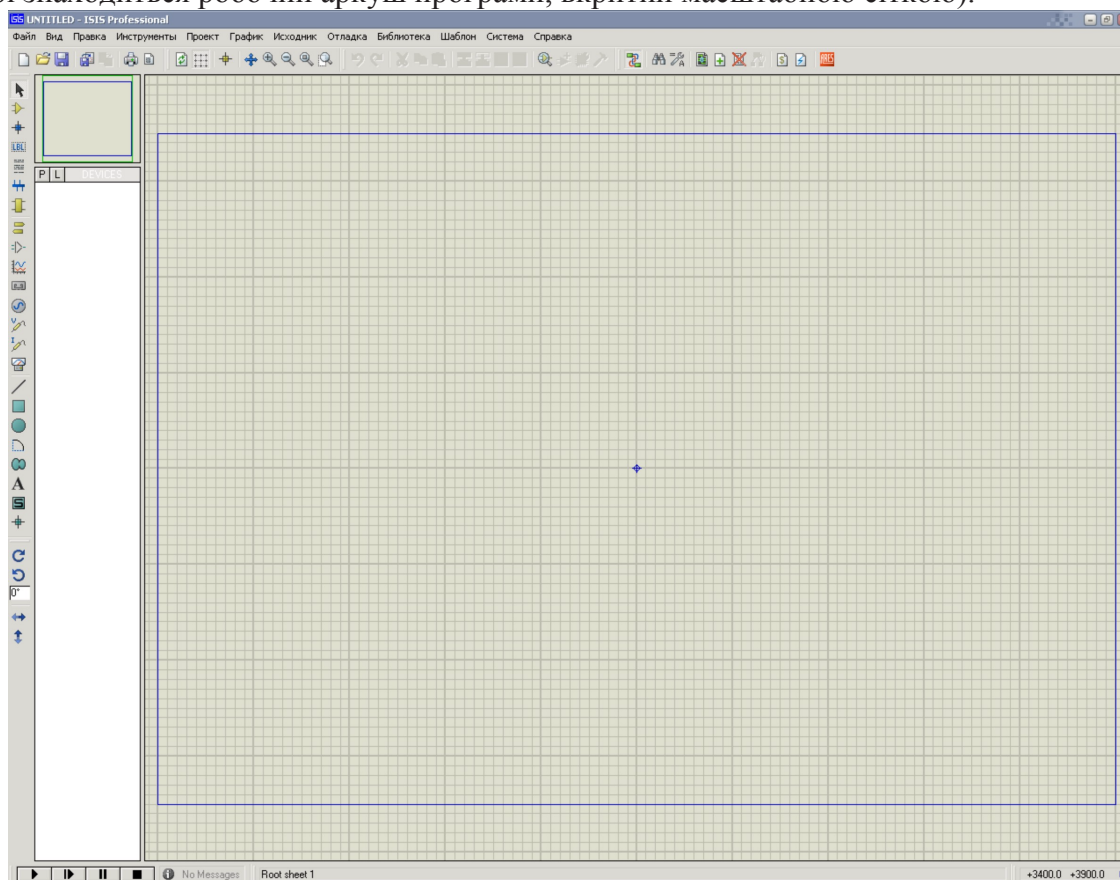
Рис. 4

Режим коливального приводу (також званий «однофазним» приводом) вибирається низьким рівнем на вході HALF/FULL, коли контролер знаходиться в парному стані (2, 4, 6 або 8)(рис. 4).

## Створення проекту з використанням контролерів L297 та L298N

Після інсталяції програми «Proteus 7» відкрити програму (двічі клацнути лівою клавiшею миші на її іконці  ).

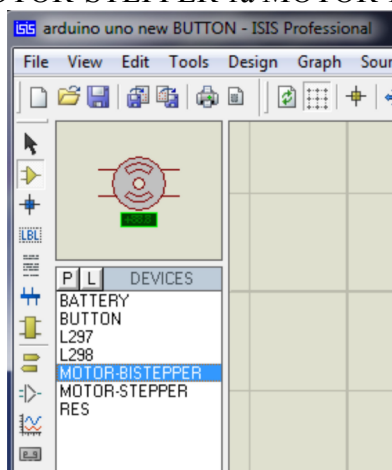
Відкриється інтерфейс користувача програми, вигляд якого зображено на рис. 1 (в центрі знаходиться робочий аркуш програми, вкритий масштабною сіткою).



**Рис. 5. Інтерфейс користувача програми «Proteus 7»**

Створимо електричну схему на основі мікросхем L297 та L298. Для цього додамо за допомогою закладки «DEVICES» на полі «Панелі інструментів», яка розташована вертикально зліва від робочого аркуша, потрібні елементи схеми:

- ◇ - батарею живлення BATTERY, напруга якої може змінюватися в межах 5...46 В,
- ◇ - кнопка BUTTON,
- ◇ - мікросхеми L297 та L298,
- ◇ - резистори RES,
- ◇ - крокові двигуни MOTOR-STEPPER та MOTOR-BISTEPPER.



**Рис. 6. Функціональні елементи для електричної схеми**

Клацнувши лівою кнопкою миші на потрібному елементі, курсором встановлюємо

його на робочий аркуш та з'єднуємо з іншими (рис. 7).

При необхідності, за наявності великої кількості елементів схеми, попередньо збільшуємо розмір робочого поля (знаходиться в середині синього прямокутника) наступним чином:

- ◇ тиснемо курсором на вкладку «System» і далі «Set Sheet Sizes» (рис. 7).
- ◇ у вікні, що відкриється, вибираємо формат робочого поля, клацнувши курсором справа від вибору.

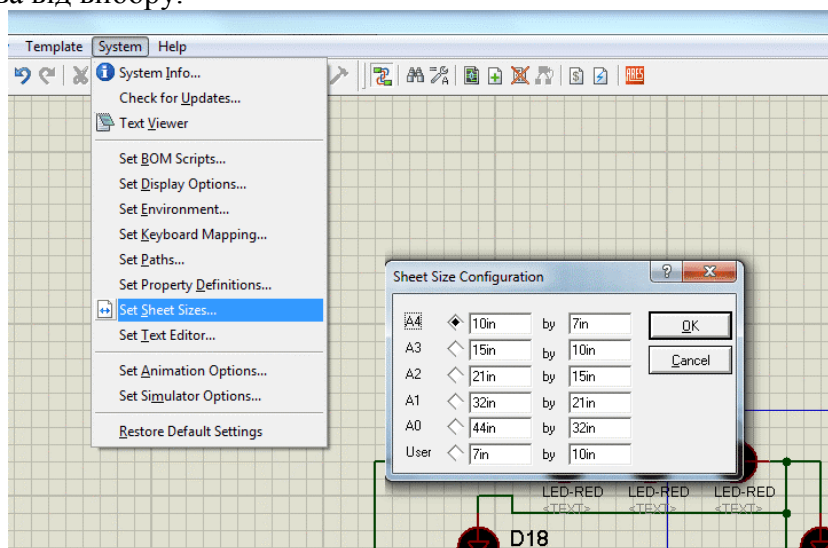


Рис. 7

Проаналізуємо роботу біполярного та уніполярного крокових двигунів.

У **біполярного** двигуна по одній обмотці в кожній фазі, тобто, всього дві обмотки і відповідно 4 виводи (рис. 8,а).

**Уніполярний** двигун також, як і біполярний, для кожної фази має по одній обмотці, але кожна обмотка містить відвід від середини. У зв'язку з цим, шляхом перемикання половинок обмоток крокового двигуна з'являється можливість змінювати напрям магнітного поля (рис. 8,б).

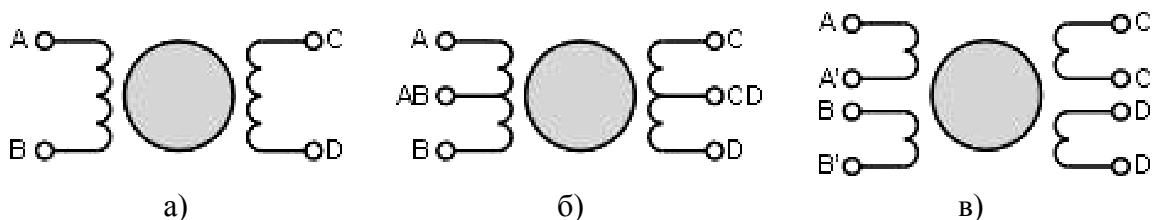
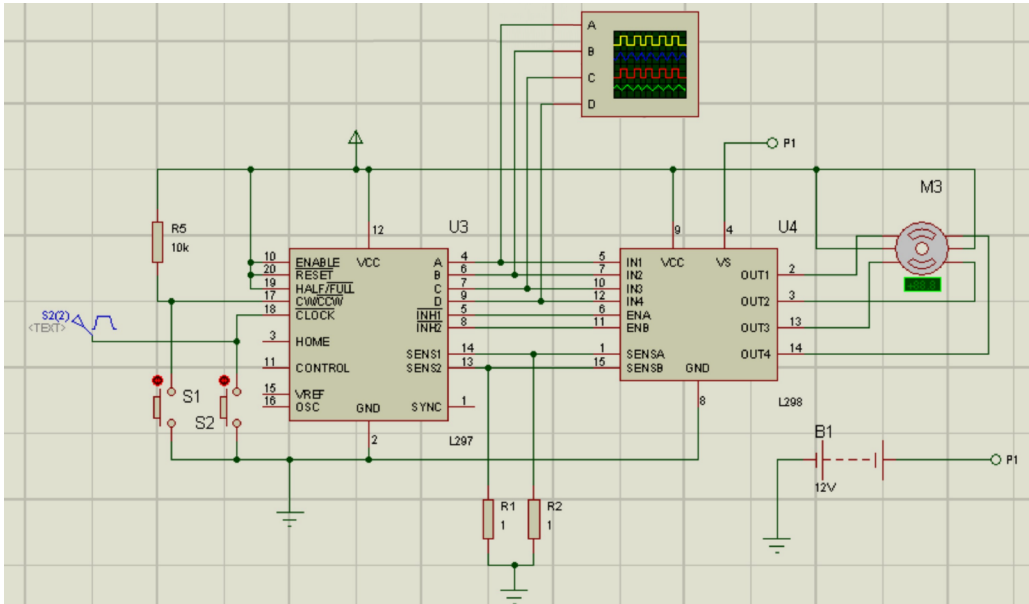


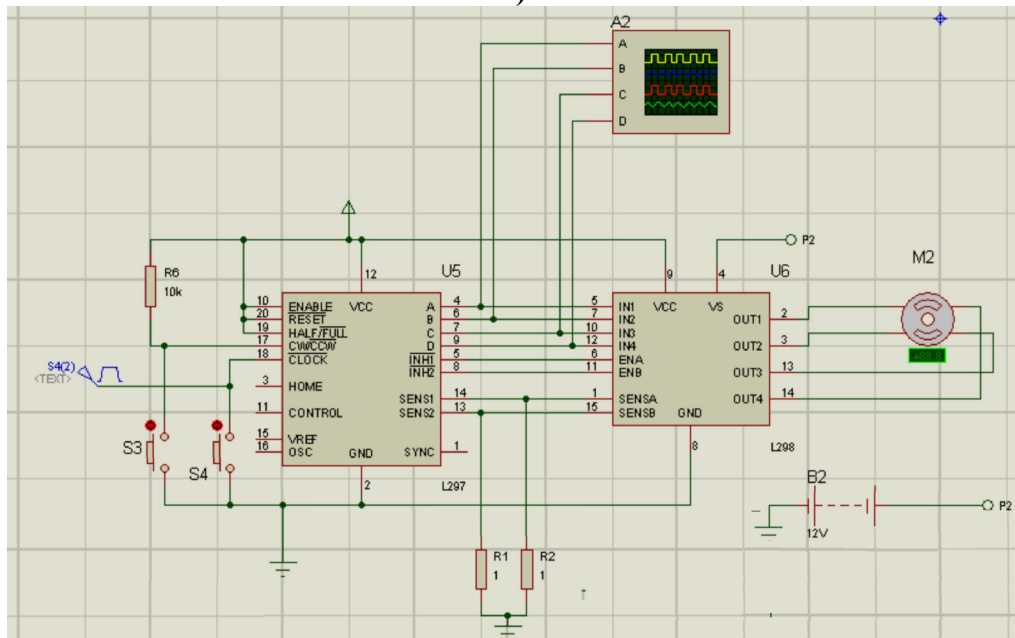
Рис. 8. Біполярні (а) та уніполярні (б) крокові двигуни

Побудуємо на основі вибраних елементів схему, яка показана на рис. 9. Схема дозволяє реалізувати два завдання - проаналізувати роботу мікросхем L297 та L298 з кроковим **уніполярним** (верхня схема на рис. 9) та кроковим **біполярним** (нижня схема на рис. 9) двигунами:

- 1) за наявності імпульсного генератора, який формує вхідні імпульси для мікроконтролера L297,
- 2) при натисканні кнопок S2 (S4), в момент відпускання яких після натискання змінюється рівень сигналів на виході L297.



а)



б)

Рис. 9

Перед аналізом роботи схеми необхідно змінити параметри генератора (на схемі рис. 9 позначені як S2(2) та S4(2)).

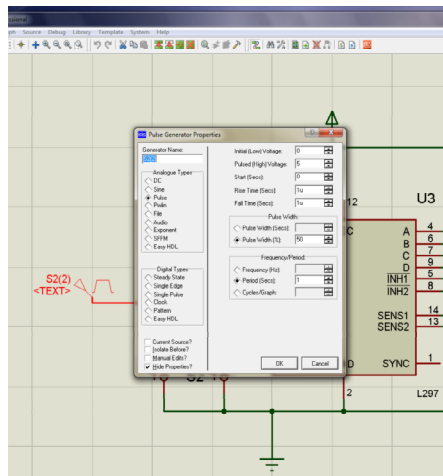


Рис. 10

Для цього двічі натисніть курсором на позначенні генератора (наприклад, на S2(2)) і у

вікні, що відкриється (рис. 10), встановить форму сигналу «Pulse», частоту «Freguency (Hz)» або період «Period (Secs)», а також рівень напруги «Pulsed High Voltage», рівний 5 В.

Окремо схема керування **уніполярним** кроковим двигуном показана на рис. 11, а схема керування **біполярним** кроковим двигуном – на рис. 12.

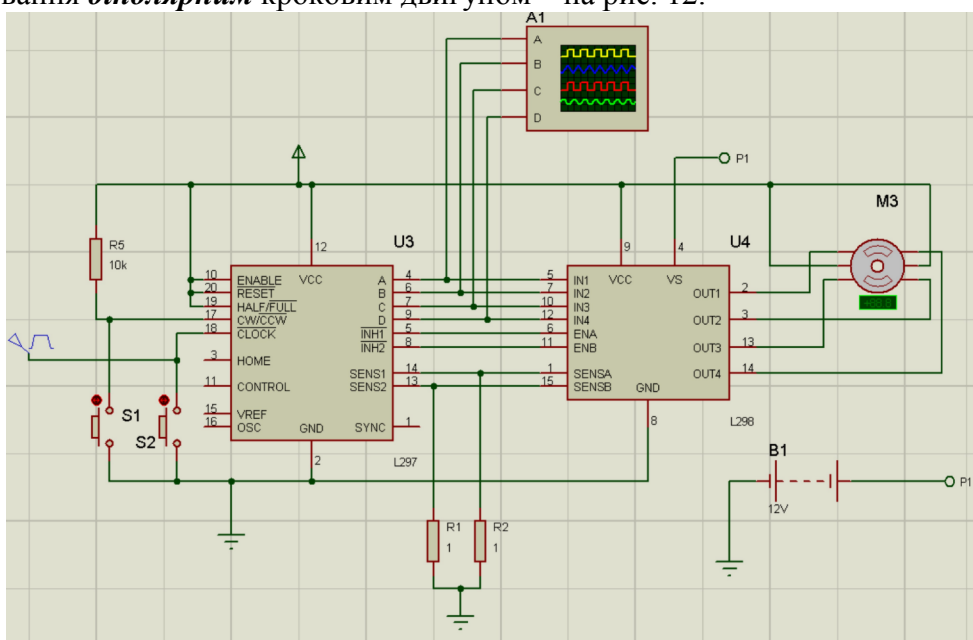


Рис. 11

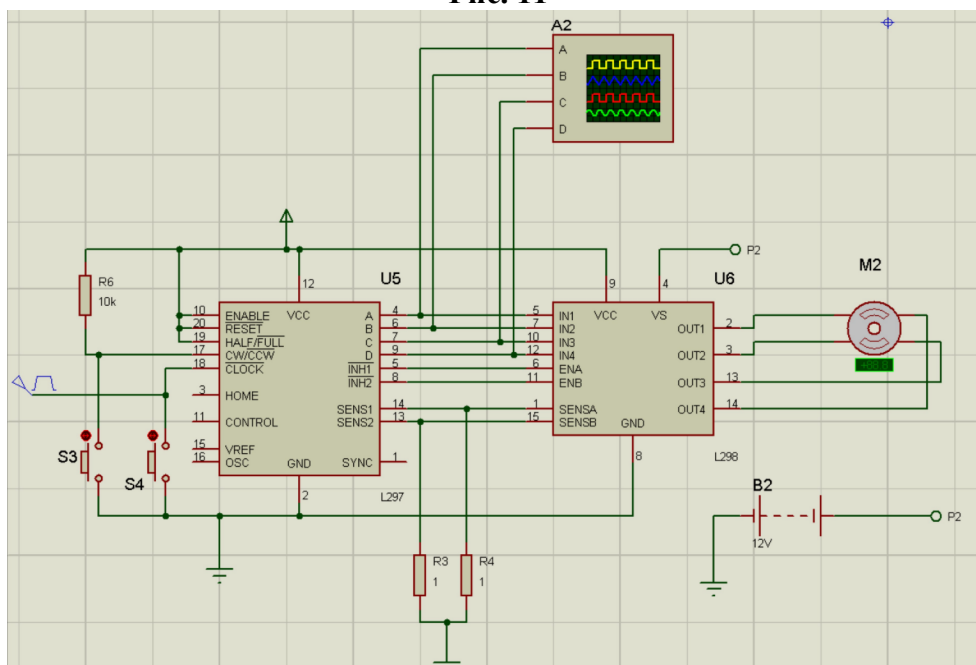
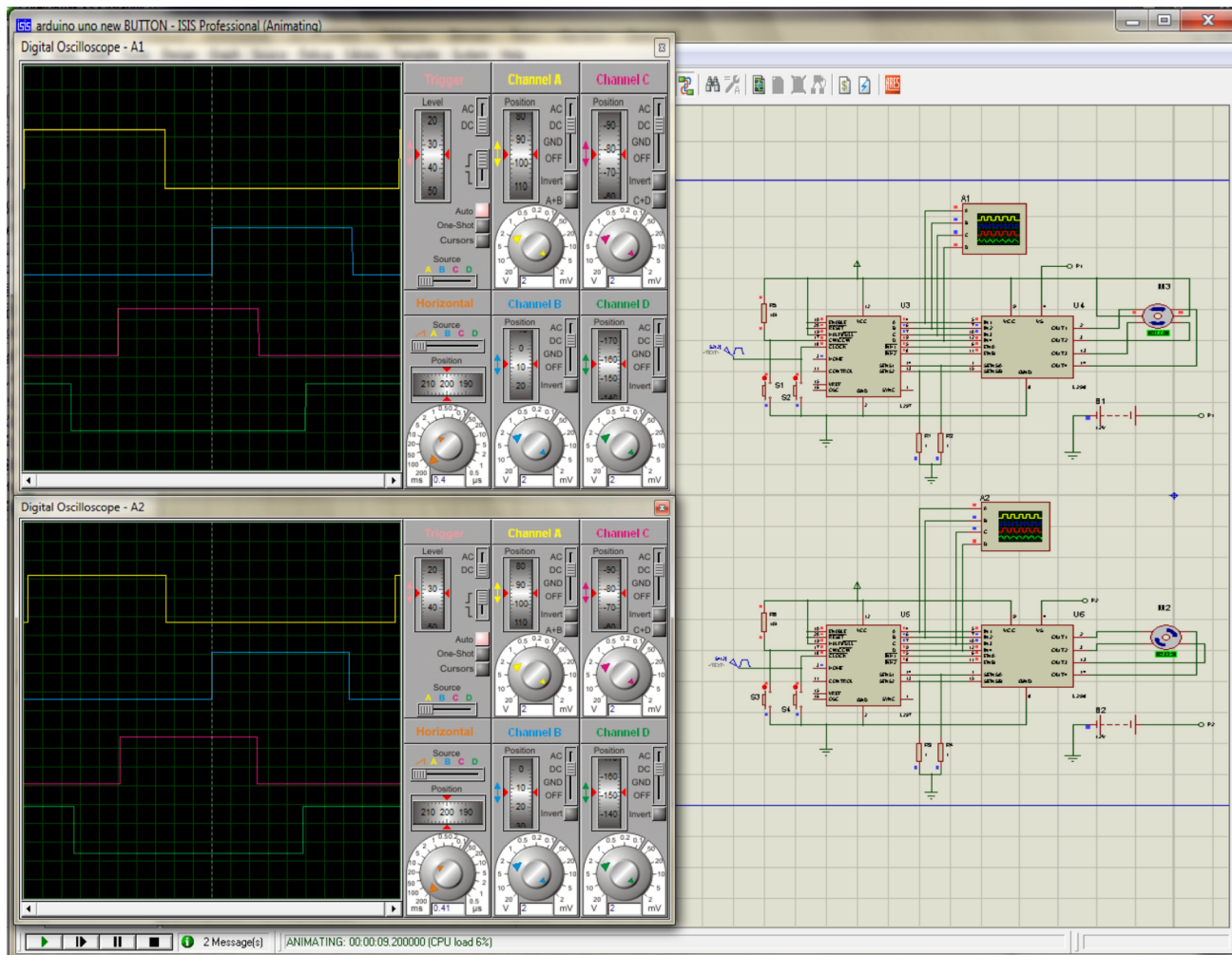


Рис. 12

Для контролю за послідовністю імпульсів, які формує контролер L297, в обох схемах до виходів мікросхеми під'єднуємо чотириканальні осцилографи. Для запуску осцилографа під час виконання проекту, якщо він не відкриється самостійно, потрібно клацнути на його зображенні правою кнопкою миші і у вікні, що розкриється, клацнути лівою кнопкою на нижньому рядку «Digital Oscilloscope».

Налаштуйте осцилограф, встановивши відповідним чином перемикачі на його панелі так, як це показано на рис. 13, клацнувши курсором на потрібних величинах розміру розгортки та масштабу амплітуди вхідного сигналу.



**Рис. 13. Налаштування осцилографів під час виконання проекту**

В обох схемах **напря́м оберта́ння** двигунів визначається положенням кнопок S1 та S3.  
**Керування кроковим двигуном з використанням схеми на основі модуля Arduino Uno**

Побудуємо електричну схему керування уніполярним кроковим двигуном на основі модуля Arduino Uno (рис. 14).

Режим роботи двигуна та частота обертавання визначається параметрами контролера. Двигун під'єднується до модуля через ключі на основі біполярних транзисторів Q1...Q4, які виконують функцію підсилювачів потужності. Транзисторні ключі увімкнені за схемою із спільним емітером. Двигун живиться від батареї B1, напруга якої перевищує напругу живлення самого модуля (на с схемі 12 В).

Для контролю за послідовністю імпульсів на виході мікроконтролера до виходів мікросхеми під'єднуємо чотириканальний осцилограф. Налаштовуємо осцилограф, встановивши відповідним чином перемикачі на його панелі так, як це показано на рис. 14, клацнувши курсором на потрібних величинах розміру розгортки та масштабу амплітуди вхідного сигналу.

Створимо для цієї схеми скетч за допомогою середовища Arduino Ide з використанням асемблерної вставки (Приклад 1). Реалізуємо повнокроковий режим роботи (рис. 3), в якому на виходи А,В,С,Д двигуна послідовно подаються наступні значення напруг:

D	C	B	A
0	1	0	1
1	0	0	1
1	0	1	0
0	1	1	0

Для забезпечення можливості регулювання частоти вихідних імпульсів мікроконтролера вводимо в скетч часову затримку імпульсів, величину якої зможемо



змінювати програмно. Враховуємо, що один робочий цикл мікроконтролером виконується за 0,5 мкс. Регістр R18 використовуємо як лічильник циклів. Підпрограму часової затримки позначимо в Прикладі 1 як «Pause».

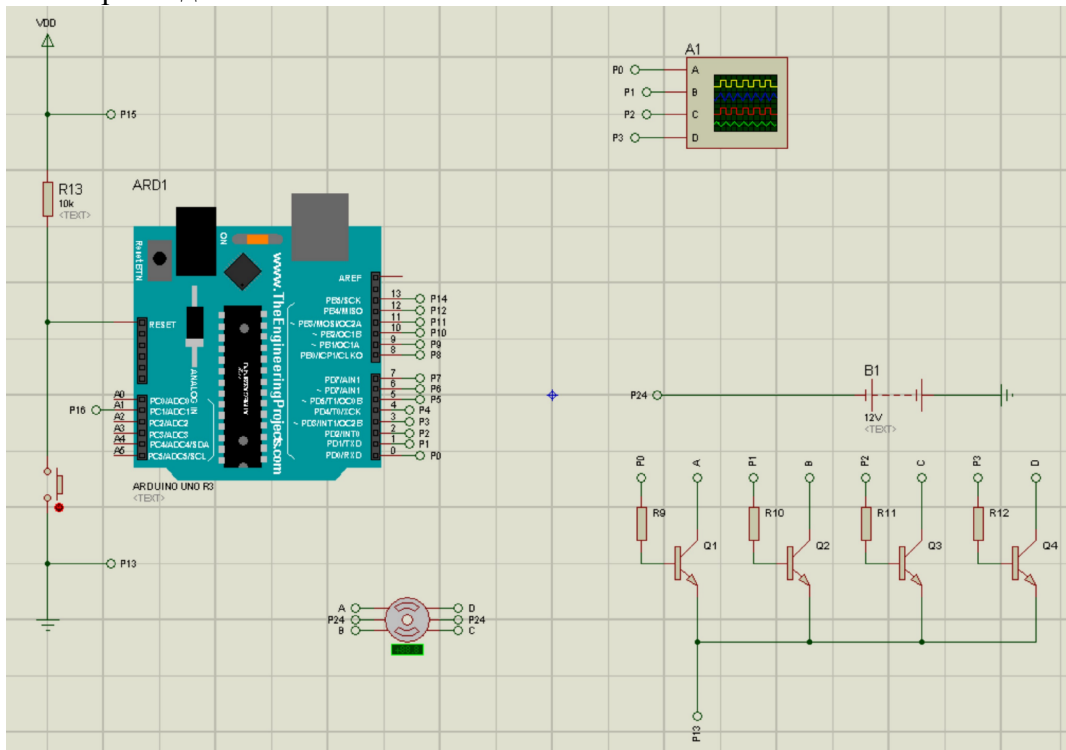


Рис. 14

### Приклад 1

```
void setup() {
    // put your setup code here, to run once:
    asm volatile
    (
        " ldi R16,0x00" "\n"
        " ldi R17,0xFF" "\n"
        " out 0x0A,R17" "\n"
        " out 0x0B,R16" "\n"
    );
}

void loop() {
    // put your main code here, to run repeatedly:
    asm volatile
    (
        " M1: " "\n"
        " ldi R18,0b11110000" "\n"
        " ldi R20,0x05" "\n"
        " out 0x0B,R20" "\n"
        " rcall Pause " "\n"
        " ldi R18,0b11110000" "\n"
        " ldi R20,0x09" "\n"
        " out 0x0B,R20" "\n"
        " rcall Pause " "\n"
        " ldi R18,0b11110000" "\n"
        " ldi R20,0x0A" "\n"
        " out 0x0B,R20" "\n"
        " rcall Pause " "\n"
        " ldi R18,0b11110000" "\n"
        " ldi R20,0x06" "\n"
    );
}
```

```

" out 0x0B,R20" "\n"
" rcall Pause " "\n"
" jmp M1" "\n"

" Pause: " "\n"
" dec R18 " "\n"
" brne Pause " "\n"
" ret " "\n"
);
}

```

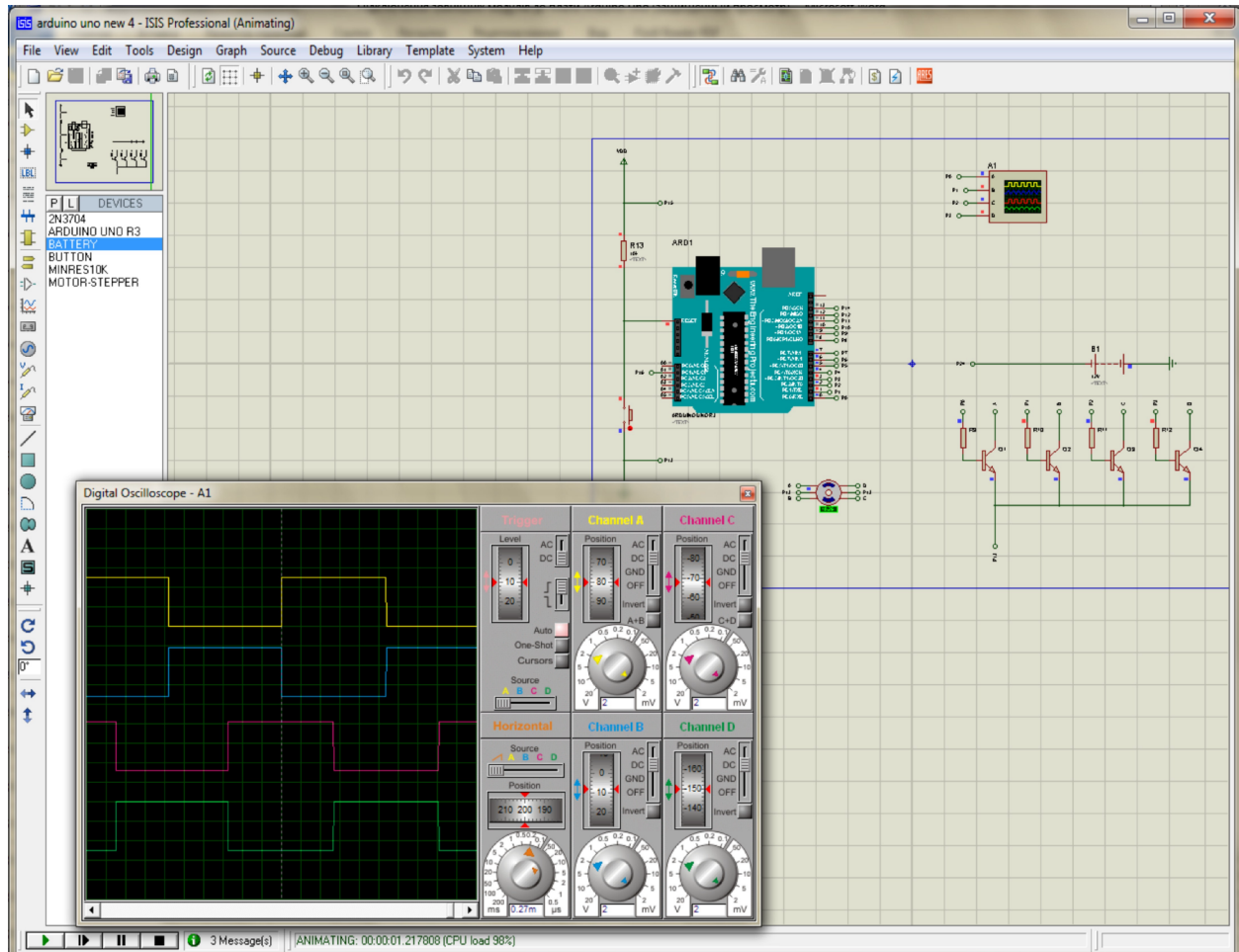


Рис. 15

Для роботи в програмному середовищі Proteus VSM, необхідно використовувати створений програмою Arduino Ide після компілювання скетчу файл з розширенням .hex.

Для цього необхідно запустити Arduino Ide і перед компілюванням скетчу зайти на вкладку «Файл» і далі «Налаштування». У вікні, що відкривається, поставити галочку навпроти опції «Показати детальний вивід: Компіляція».

Далі компілюємо скетч, натиснувши курсором на вкладці «Перевірити».

Після компілювання скетчу відкриваємо вкладку «Скетч», і у вікні, що розкриється, тиснемо лівою кнопкою миші на вкладці «Експорт бінарного файлу»;

Після цього знову тиснемо на вкладці «Скетч» і у вікні, що з'явиться, натискаємо на вкладку «Показати папку скетчу».

Розкриється вікно з папкою скетчу, в якому, крім робочого файлу скетчу, будуть ще два файли з розширенням .hex. Копіюємо перший з них і записуємо його у вибрану папку. Це і є файл, який буде потрібен для подальшої роботи з Proteus VSM.

Запускаємо виконання проекту натисканням лівою клавішею миші кнопку «Play» **Ошибка! Объект не может быть создан из кодов полей редактирования.** на панелі, яка розташована внизу зліва від робочого поля (аркуша). Після запуску проекту на виконання значок на кнопці змінить колір з чорного на зелений (рис. 15).



## **Виконання лабораторної роботи**

### **Завдання № 1**

1. Побудуйте схему електричну принципову відповідно до рис. 9 з використанням програмного середовища Proteus VSM. Збережіть робочий проект.

1.1. Встановіть наступні параметри генераторів S2(2) та S4(2): виберіть форму сигналу «Pulse», період сигналу «Period (Secs)», рівний 1 секунді, а також рівень напруги «Pulsed High Voltage», рівний 5 В.

2. Запустіть проект на виконання.

3. Налаштуйте осцилографи, встановивши відповідним чином перемикачі на їх панелі так, як це показано на рис. 13. При правильних розмірах розгортки осцилографів отримаєте форму сигналів на всіх чотирьох виводах А,В,С,Д мікросхеми L297.

4. Зробіть «Print Screen» екрану монітора з результатом виконання лабораторної роботи і додайте його у звіт.

### **Завдання № 2**

1. В створеному проекті (рис. 9) видаліть генератори (клацнути правою кнопкою на зображенні генератора і у вікні, що розкриється, вибрати опцію «Delete Object»).

2. Запустіть проект на виконання.

3. Натискаючи по чергово кнопки S2 (для верхньої схеми) та S4 (для нижньої) можна змусити двигуни обертатися за часовою стрілкою. Одне натискання кнопки – один крок (півкрок) двигуна.

4. Зробіть «Print Screen» екрану монітора з результатом виконання лабораторної роботи і додайте його у звіт.

5. Зупиніть виконання проекту і натисніть кнопки S1 та S3, замкнувши їх.

6. Запустіть проект на виконання. Зробіть «Print Screen» екрану монітора з результатом виконання лабораторної роботи. Додайте його у звіт.

### **Завдання № 3**

1. Реалізуйте півкроковий режим роботи двигуна (рис. 2), використовуючи як зразок скетч, приведений в Прикладі 1.

2. Створіть скетч і після його компілювання скопіюйте файл з розширенням .hex в вибрану папку.

3. Запустіть виконання проекту, використовуючи отриманий файл.

4. Зробіть «Print Screen» екрану монітора з результатом виконання лабораторної роботи і додайте його у звіт.

5. Зменшіть частоту обертання двигуна в 10 разів, змінивши параметри часової затримки в скетчі. Скопіюйте скетч. Скопіюйте файл з розширенням .hex і завантажте його в модуль Arduino Uno.

6. Запустіть виконання проекту, використовуючи отриманий файл.

7. Зробіть «Print Screen» екрану монітора з результатом виконання лабораторної роботи і додайте його у звіт.

### **Література**

1. Мікропроцесорні системи управління: навч. посіб./ В.О.Денисюк, С.М.Цирульник; Вінн. нац. аграр. ун-т. Вінниця: ТВОРИ, 2021. 204.

2. Методичні вказівки до лабораторних робіт з дисципліни «Електротехніка та електроніка» для студентів спеціальності 122 Комп'ютерні науки і інформаційні технології» денної форм навчання / Укл.: А.В. Пархоменко, О.М. Гладкова. – Запоріжжя: ЗНТУ, 2016. – 41 с.

3. Розробка програмних модулів для обміну даними у промислових мережах: [Електронний ресурс] : навч. посіб. для студ. спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» / Укладачі: А. В. Сагун, В. В. Хайдуров, І. А. Поліщук; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 16,2 МБайт). – Київ : КПІ ім. Ігоря Сікорського, 2021. – 103 с.

4. L297.STEPPER MOTOR CONTROLLERS. Datasheet.\www.datasheetcatalog.com.

## Зміст

Використання контролерів L297 та L298N для керування кроковими двигунами .....	3
Призначення виводів L297.....	3
Послідовність фаз руху двигуна .....	4
Півкроковий режим роботи двигуна .....	5
Повнокроковий режим роботи двигуна .....	5
Коливальний режим роботи .....	5
Створення проекту з використанням контролерів L297 та L298N.....	6
Керування кроковим двигуном з використанням схеми на основі модуля Arduino Uno .....	10
Виконання лабораторної роботи.....	13
Завдання № 1.....	13
Завдання № 2.....	13
Завдання № 3.....	13
Література .....	13