

**Міністерство освіти і науки України  
Національний технічний університет  
«Дніпровська політехніка»**



**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЕЛЕКТРОЕНЕРГЕТИКИ  
ЕЛЕКТРОТЕХНІЧНИЙ ФАКУЛЬТЕТ  
Кафедра електропривода**

**МЕТОДИЧНІ РЕКОМЕНДАЦІЇ  
ДО ЛАБОРАТОРНИХ РОБІТ  
З ДИСЦИПЛІНИ  
«СЕРВОПРИВОД»**

спеціальності 141 «Електроенергетика, електротехніка та електромеханіка»

Дніпро  
НТУ «ДП»  
2021

Худолій С.С. та ін. Методичні рекомендації до лабораторних робіт з дисципліни «Сервоприводи» для студентів спеціальності 141 Електроенергетика, електротехніка та електромеханіка / Нац. техн. ун-т «Дніпровська політехніка». – Д. : НТУ «ДП», 2021. – 39 с.

Укладачі:

С.С. Худолій, канд. техн. наук, проф. ;  
О.С. Бешта, д-р техн. наук, проф. ;  
О.О. Азюковський, канд. техн. наук, проф. ;  
О.В. Балахонцев, канд. техн. наук, доц. ;  
Р.О. Боровик, асистент.

Погоджено рішенням методичної комісії спеціальності 141 «Електроенергетика, електротехніка та електромеханіка» (протокол № 20\21-05 від 10.03.2021).

Методичні вказівки призначені для студентів спеціальності 141 «Електроенергетика, електротехніка та електромеханіка».

## 1. ЗАГАЛЬНІ ВІДОМОСТІ

Лабораторні роботи студентів 141 спеціальності є невід’ємною складовою у системі підготовки фахівців вищого навчального закладу. Лабораторні завдання спрямовані розвиток уявлення про структури та компонентів сервоприводів, розрізняти переваги та недоліки різних приводів та області їх застосування, розуміти принцип дії та призначення компонентів сервоприводу, розуміти схеми підключень сервоприводу та вміти здійснювати підключення компонентів сервоприводу, розуміти алгоритми налагодження сервоприводу та вміти налаштувати сервоприводи.

Шляхом експериментальної перевірки на лабораторних роботах досліджуються теоретичні основи роботи та керування сервоприводами різних типів.

Структура курсу направлена на виконання реальних задач фахівця з електромеханіки: зрозуміти задачу позиціонування технологічного об’єкту, виконати правильні розрахунки параметрів сервоприводу, обрати необхідний тип сервоприводу та окремих компонентів, врахувати фактори, що впливають на якість керування сервоприводом, успішно підключити та налагодити електромеханічну систему з використанням сервоприводу

Основний навчальний час приділяється виконанню практичних завдань, об’єктному програмуванню, моделюванню роботи електромеханічної системи, підготовки звіту.

Ці методичні вказівки мають полегшити студентам пошук вихідних даних для виконання завдань та допомогти їм цілеспрямовано та якісно виконати поставлені завдання.

## 2. ВИМОГИ ДО ОФОРМЛЕННЯ ЗВІТУ З ЛАБОРАТОРНИХ РОБІТ

Кожна лабораторна робота оформлюється окремо, подається на перевірку у електронному вигляді (формат MS Word) шляхом надсилання на електронну пошту викладача. Перевірені роботи роздруковуються та вкладаються до особистої папки з лабораторних робіт. Після виконання всіх лабораторних робіт папка з звітами подається викладачу.

### ЗМІСТ ЗВІТУ

1. Титульний лист – один для всього звіту.
2. Номер та тема лабораторної роботи.
3. Мета та завдання до лабораторної роботи.
4. Розроблена схема керування сервоприводом.
5. Програмний код (скетч) з коментарями до кожного рядка.
6. Графіки вихідних сигналів з коментарями.
7. Висновки до лабораторної роботи.
8. Результати завдань до самостійного виконання.
9. Перелік використаних інформаційних джерел.

Оформлення звіту здійснюється відповідно до ДСТУ 3008:2015. Звіти у сфері науки і техніки. Структура та правила оформлення. Мова звіту – українська.

Текст записки виконують на аркушах білого паперу формату А4 (210 x 297 мм) за допомогою комп'ютера (розмір шрифту 14 пт.). Текст розміщується на одній стороні аркушу. Відстані від рамки аркуша до межі тексту: зверху та знизу 15 мм, з лівого та правого боків 5 мм. Абзацний відступ 15 мм.

Нумерація аркушів (включно з рисунками та таблицями) – незалежна в межах усієї записки. Титульний аркуш та завдання не нумеруються, але в підрахунках загальної кількості аркушів записки враховуються.

У тексті пояснювальної записки наводяться в разі потреби рисунки, схеми та таблиці. Нумерація рисунків та таблиць повинна бути двоступінчастою (подібно до нумерації підрозділів). Таблиці повинні мати заголовки, рисунки – підрисункові підписи, які пояснюють зміст рисунку або таблиці.

Посилання на літературні джерела даються у прямокутних дужках, де позначається номер відповідного джерела у переліку посилань (наприклад, так: [4]).

Розрахунково-пояснювальна записка повинна бути написана з правильним використанням технічної термінології, визначень та загальноживаних умовних буквених позначень фізичних та математичних величин. Усі розрахунки виконуються в системі СІ.

### 3. ЗАВДАННЯ ДО ЛАБОРАТОРНИХ РОБІТ

**Лабораторна робота №1.** Дослідження роботи сервоприводів Micro Servo 9g SG90 з системою керування на базі Arduino.

**Мета роботи** – дослідити роботу сервоприводу типу Micro Servo 9g SG90 з використанням системи керування на базі платформи Arduino Uno.

**Постановка задачі** – виконати підключення основних елементів електромеханічної системи та розробити програмний код для системи керування.

#### Порядок виконання роботи

1. Вибрати необхідні елементи та з'єднати їх згідно схеми підключення (рис.3.1)

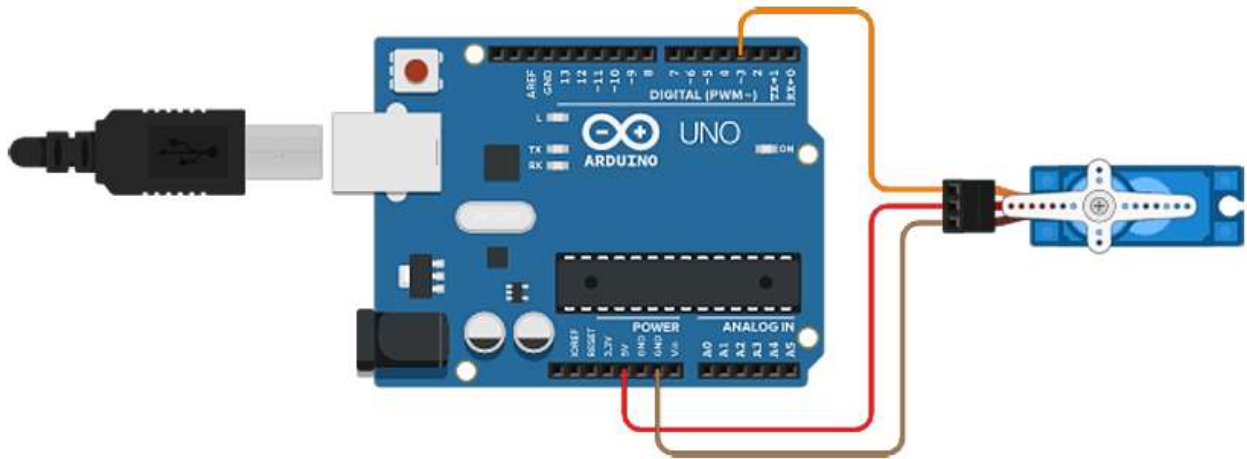


Рис.3.1 – Схема підключення сервоприводу до плати керування.

2. Розробити програмний код системи керування для забезпечення повороту валу сервопривода згідно діаграм (рис.3.2).

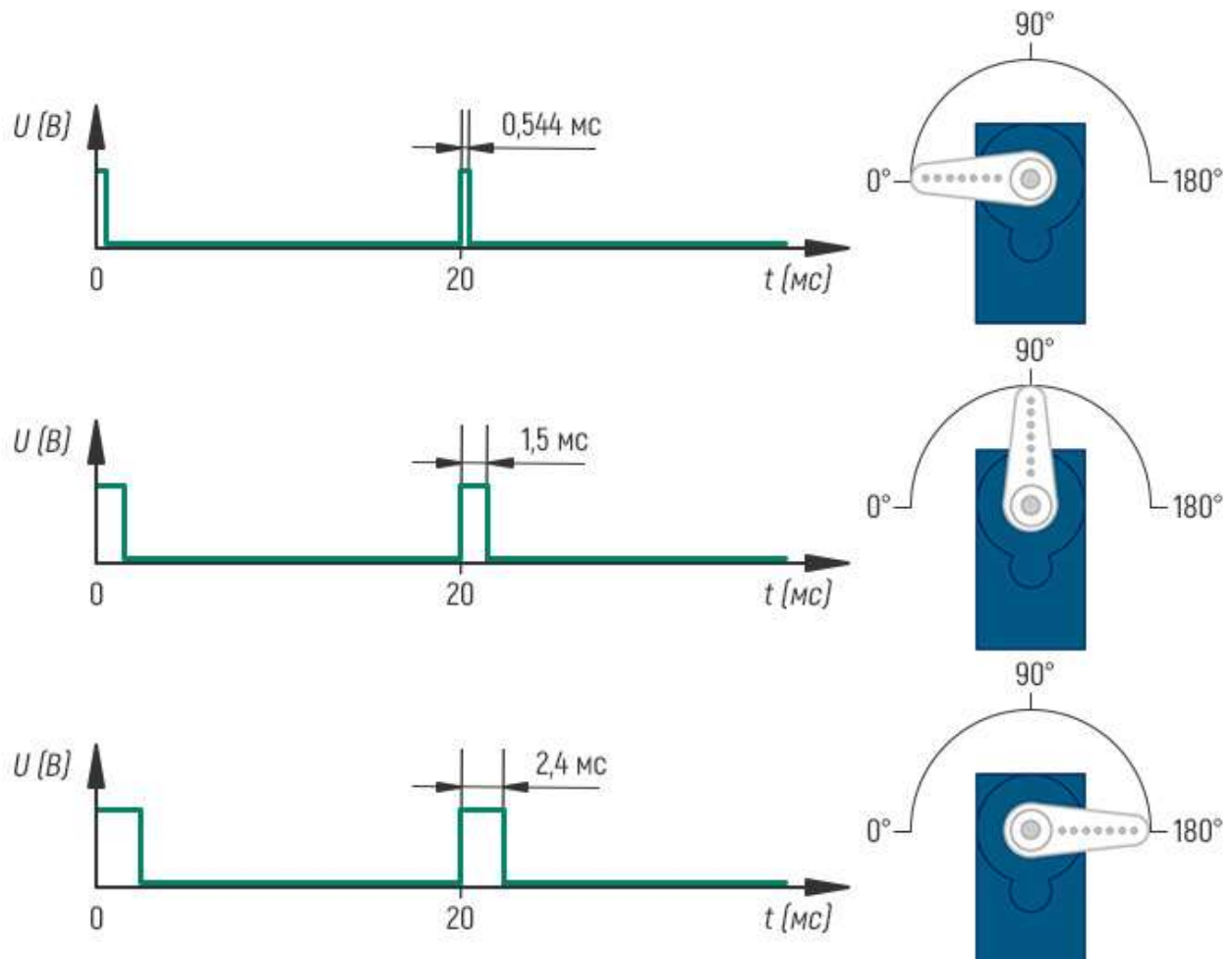


Рис.3.2 – Діаграма сигналів керування сервоприводом.

**Приклад програмного коду:**

```
#include <Servo.h> //використовуємо бібліотеку для роботи з сервоприводом
Servo servo; //оголошуємо змінну servo типу Servo
void setup() //процедура setup
{
servo.attach(10); //вказуємо канал підключення сервопривода до порту 10
}
void loop() //процедура loop
{
servo.write(0); //встановлення валу мотора в позицію 0
delay(2000); //затримка 2 секунди
servo.write(90); //встановлення валу мотора в позицію 90
delay(2000); //затримка 2 секунди
servo.write(180); //встановлення валу мотора в позицію 180
delay(2000); //затримка 2 секунди
```

}

3. Підключити плату керування Arduino Uno до ПК, (рис.3.3).

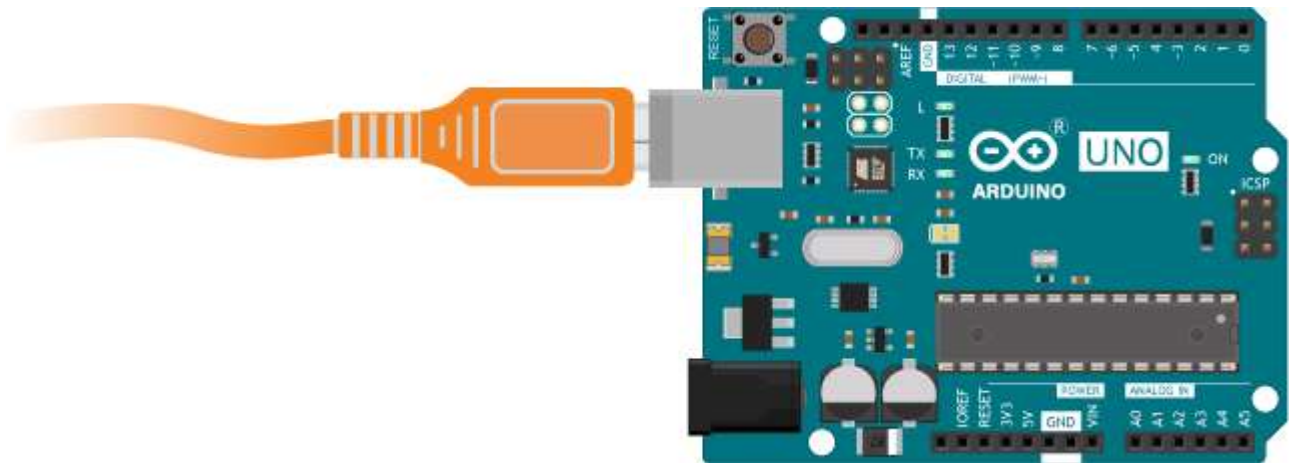


Рис.3.3 – Схема підключення плати керування Arduino Uno до ПК.

4. Провести попередню перевірку та компіляцію розробленого програмного коду у середовищі Arduino IDE (рис.3.4)

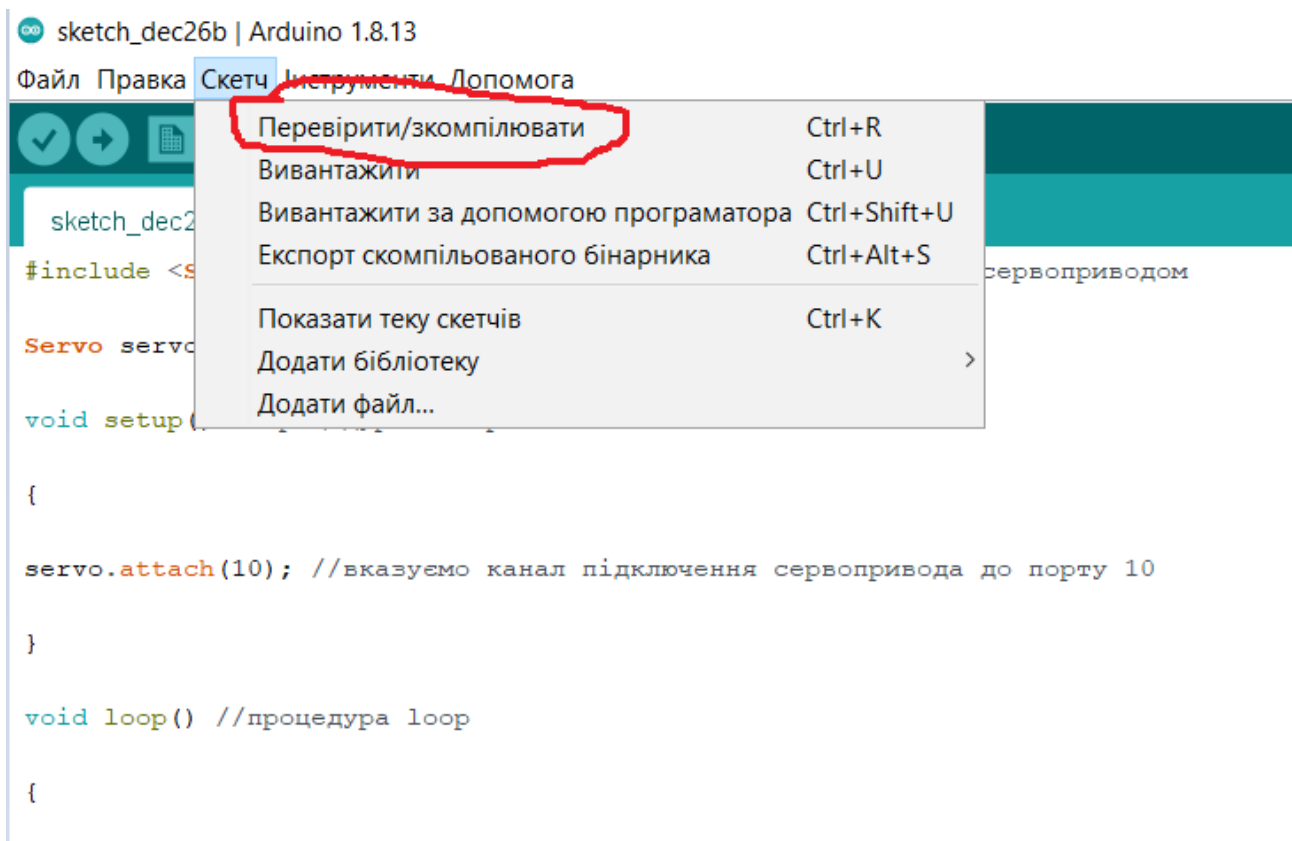


Рис.3.4 – Перевірка та компіляція програмного коду в Arduino IDE.

5. Завантажити розроблений програмний код до плати керування з середовища Arduino IDE (рис.3.5)

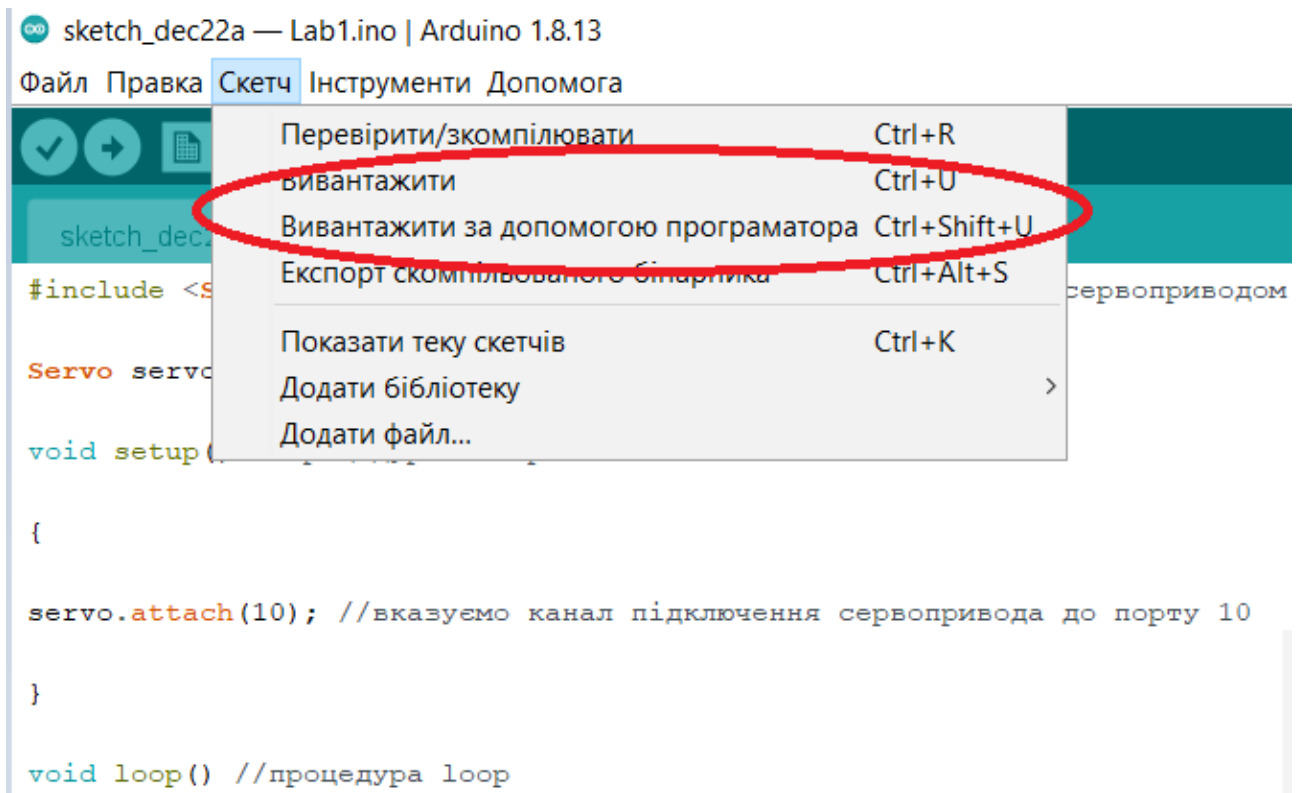


Рис.3.5 – Завантаження програмного коду з Arduino IDE до плати керування.

6. Після завантаження програмного коду, перевірити правильність виконання поставленої задачі на кут повороту вихідного валу.

7. За допомогою функції реєстрації сигналу керування «Послідовний плотер» (рис.3.6) сформувати графіки сигналів керування сервоприводом.



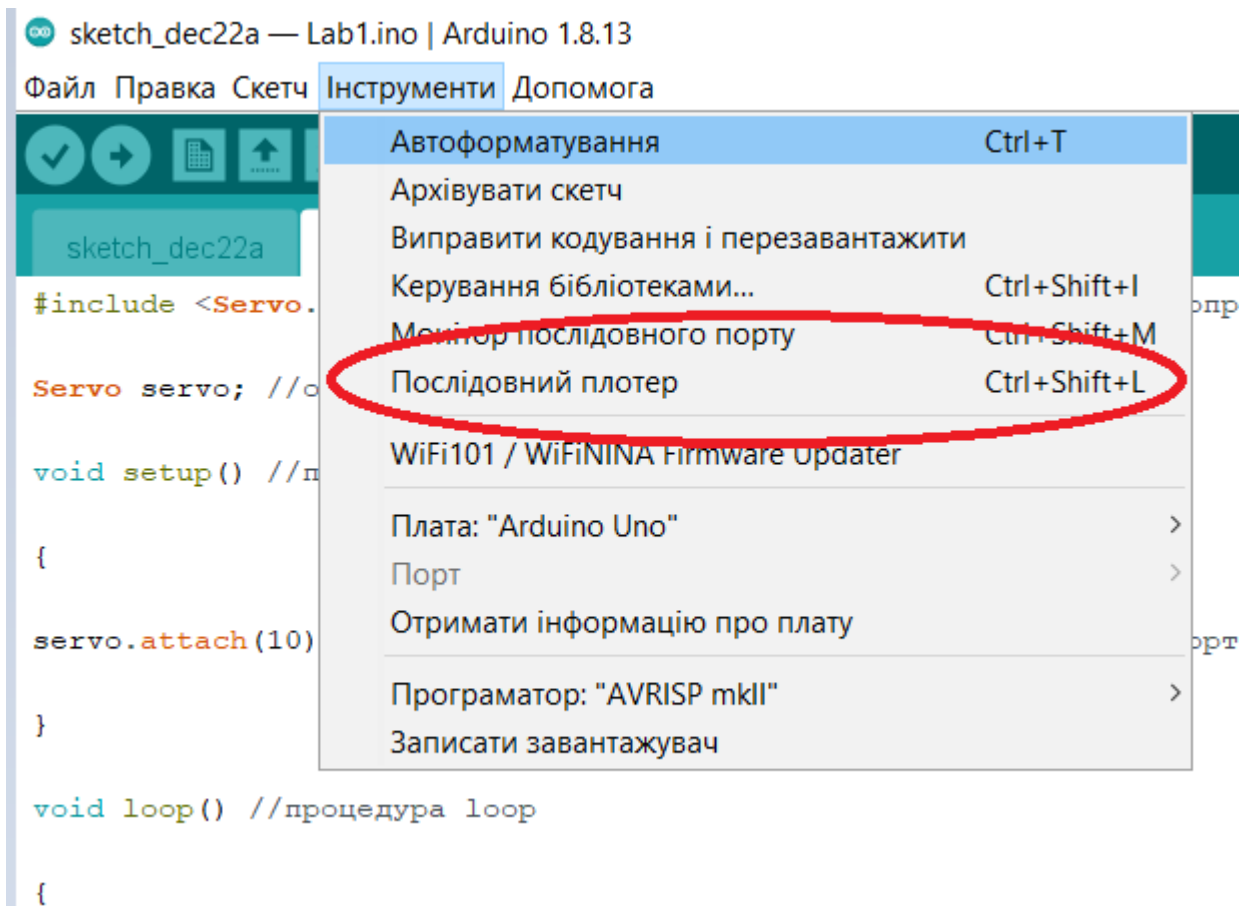


Рис.3.6 – Реєстрація вихідного сигналу в Arduino IDE за допомогою послідовного плотеру.

8. Підготувати звіт з лабораторної роботи.

### Зміст звіту з лабораторної роботи.

1. Мета та завдання до лабораторної роботи.
2. Розроблена схема керування сервоприводом.
3. Програмний код (скетч) з коментарями до кожного рядка.
4. Графіки вихідних сигналів з коментарями.
5. Висновки до лабораторної роботи.
6. Результати завдань до самостійного виконання.

### Завдання для самостійного виконання.

1. Дослідити роботу сервопривода при наявності сигналу керування від зовнішнього потенціометра (рис.3.7)

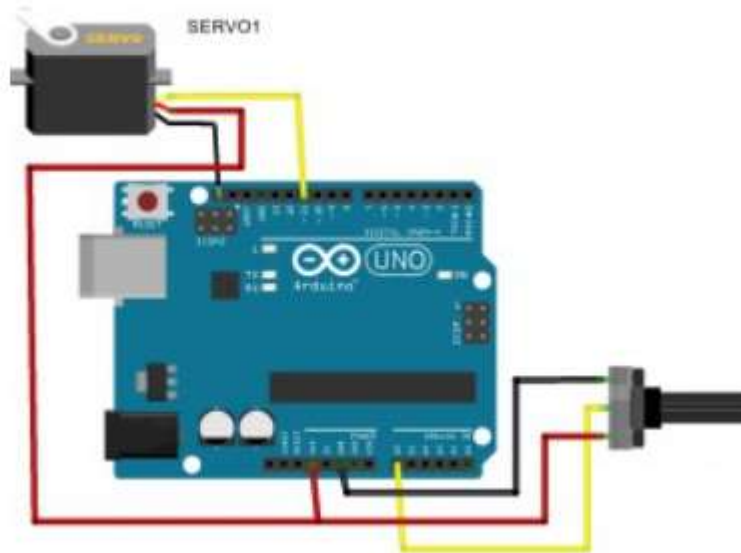


Рис.3.7 – Керування сервоприводом за допомогою зовнішнього потенціометра.

2. Розробити закон керування сервоприводом для забезпечення повороту вихідного валу в діапазоні від  $0^{\circ}$  до  $120^{\circ}$  з кроком  $10^{\circ}$  та затримкою часу між відділками в 1 сек. Дослідити роботу сервопривода при наявності такого сигналу керування.

## Лабораторна робота №2. Паралельна робота сервоприводів Micro Servo 9g SG90 з системою керування на базі Arduino.

**Мета роботи** – дослідити паралельну роботу двох і більше сервоприводів типу Micro Servo 9g SG90 з використанням системи керування на базі платформи Arduino Uno.

**Постановка задачі** – виконати підключення основних елементів електромеханічної системи та розробити програмний код для системи керування.

### Порядок виконання роботи

1. Вибрати необхідні елементи та з'єднати їх згідно схеми підключення (рис.3.8)

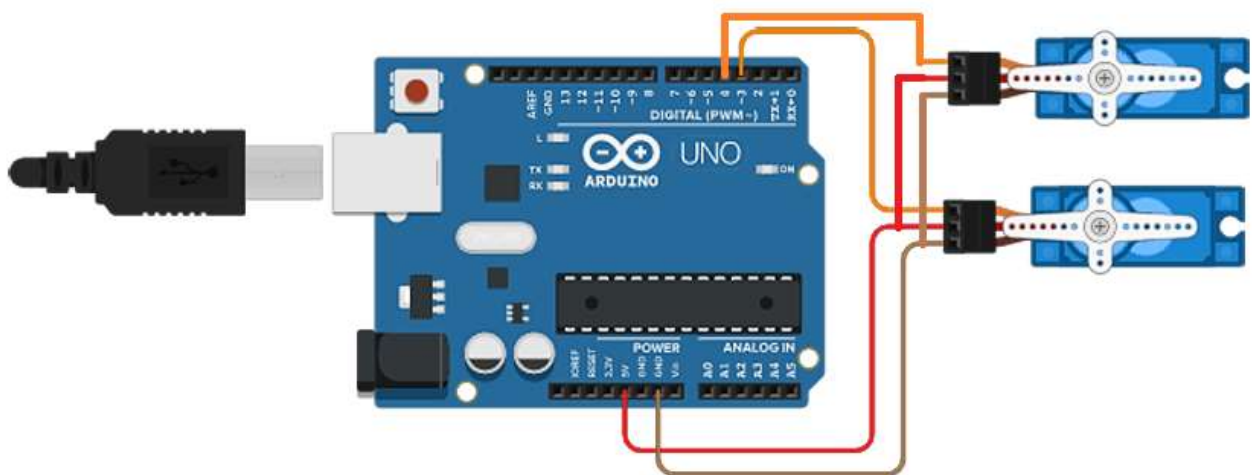


Рис.3.8 – Схема підключення сервоприводів до плати керування.

2. Розробити програмний код системи керування для забезпечення повороту валу сервопривода згідно діаграм (рис.3.9).

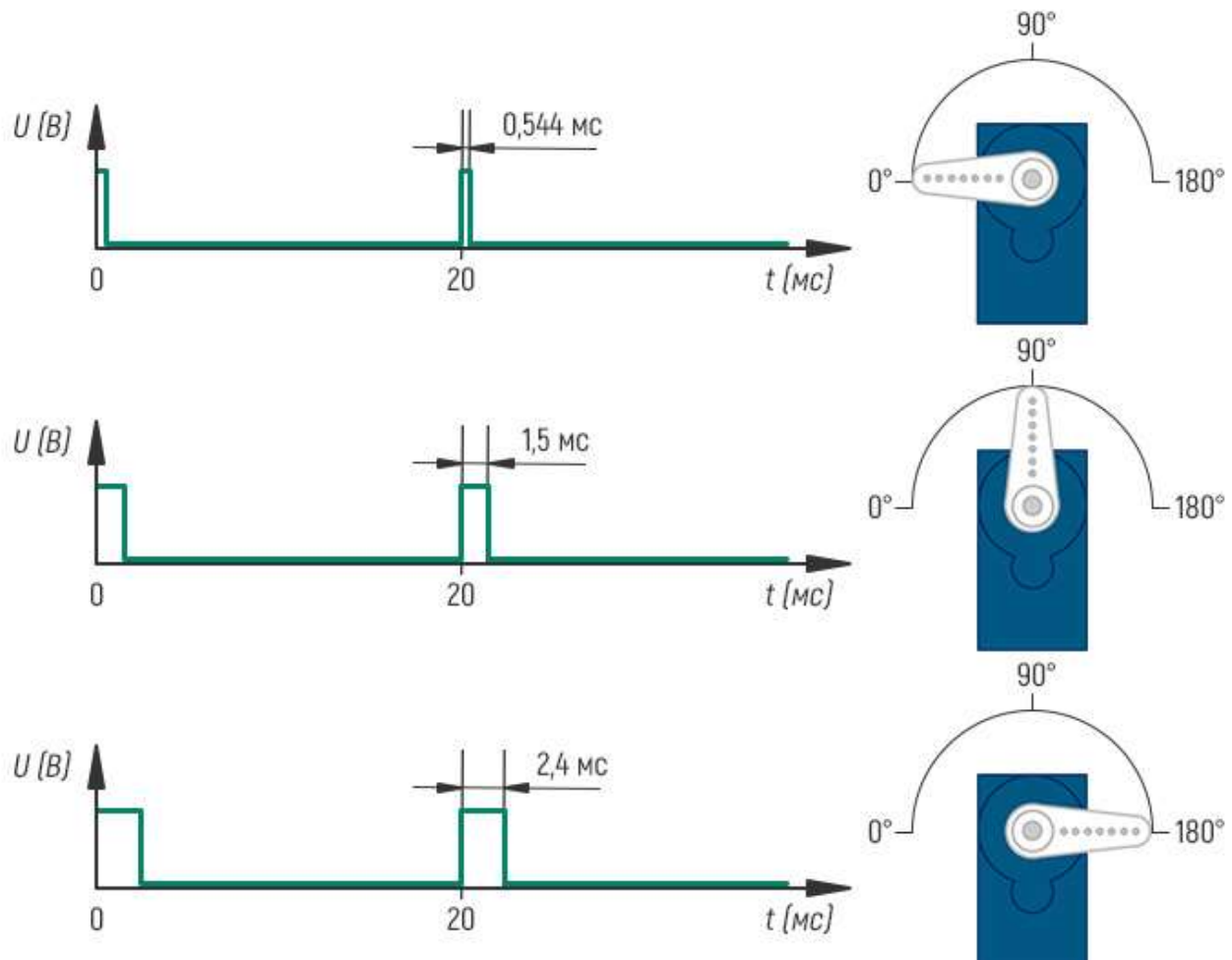


Рис.3.9 – Діаграма сигналів керування сервоприводом.

**Приклад програмного коду:**

```
#include <servo.h> //використовуємо бібліотеку для роботи з сервоприводом
Servo servo1;     // Створення першого об'єкту типу «сервопривод»

int angle1 = 0;   // змінна, місце зберігання положення першого сервопривода

int interval1 = 10; // Інтервал часу, через який здійснюється повертання першого сер-
вопривода на один градус

unsigned long lastTimeCheck1; // час останнього руху першого сервопривода, в мс від по-
чатку роботи програми

Servo servo2;     // Створення другого об'єкту типу «сервопривод»

int angle2 = 0;   // змінна, місце зберігання положення першого сервопривода

int interval2 = 20; // Інтервал часу, через який здійснюється повертання другого сер-
вопривода на один градус

unsigned long lastTimeCheck2; // час останнього руху першого сервопривода, в мс від по-
чатку роботи програми

void setup()
{
  servo1.attach(2); // вказуємо канал підключення першого сервопривода до порту 2
  servo2.attach(3); // вказуємо канал підключення другого сервопривода до порту 3
```

```

lastTimeCheck1=millis(); // Відлік часу для початку руху сервоприводів з початку роботи програми
lastTimeCheck2=millis(); // Відлік часу для початку руху сервоприводів з початку роботи програми
}
void loop()
{
  if (angle1 < 180) // Перевірка досягнення крайнього положення
  {
    if ((millis()-lastTimeCheck1)>= interval1)// Перевірка часу з останньої дії і порівняння з заданим інтервалом.
    //Якщо часу пройшло більше -виконуємо наступну дію
    {
      angle1++; // Зміна кута на один градус
      servo1.write(angle1); // Надсилання команди першому сервоприводу з новим кутом повороту
      lastTimeCheck1= millis(); // Оновлюємо таймер часу, для запуску наступного циклу з поточного моменту
    }
  }
  if (angle2 < 180) // Перевірка досягнення крайнього положення
  {
    if ((millis()-lastTimeCheck2)>= interval2) // Перевірка часу з останньої дії і порівняння з заданим інтервалом.
    //Якщо часу пройшло більше -виконуємо наступну дію
    {
      angle2++; // Зміна кута на один градус
      servo1.write(angle2); // Надсилання команди другому сервоприводу з новим кутом повороту
      lastTimeCheck2= millis(); // Оновлюємо таймер часу, для запуску наступного циклу з поточного моменту
    }
  }
}
}
}

```

3. Підключити плату керування Arduino Uno до ПК, (рис.3.10).

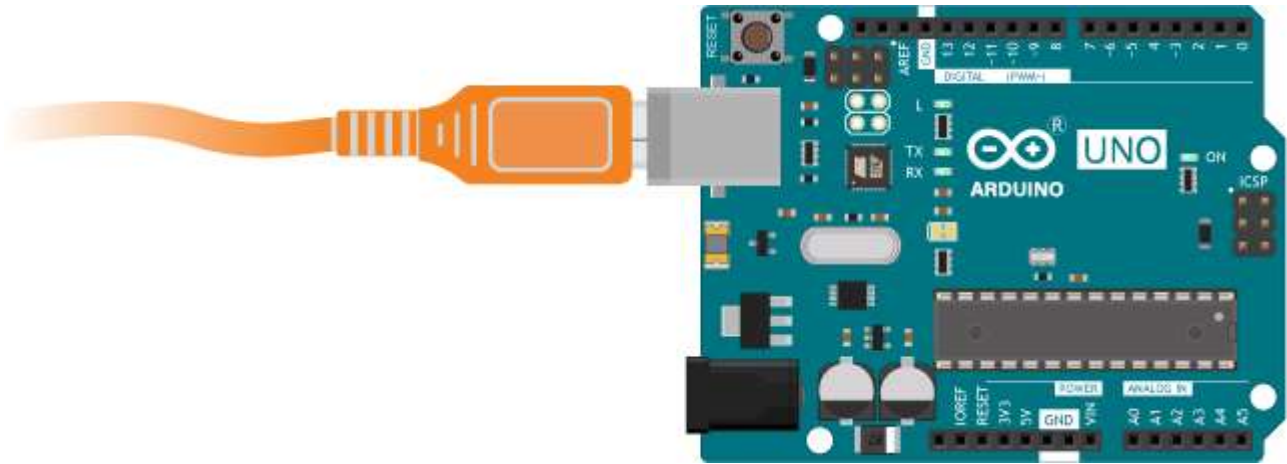


Рис.3.10 – Схема підключення плати керування Arduino Uno до ПК.

4. Провести попередню перевірку та компіляцію розробленого програмного коду у середовищі Arduino IDE (рис.3.11)

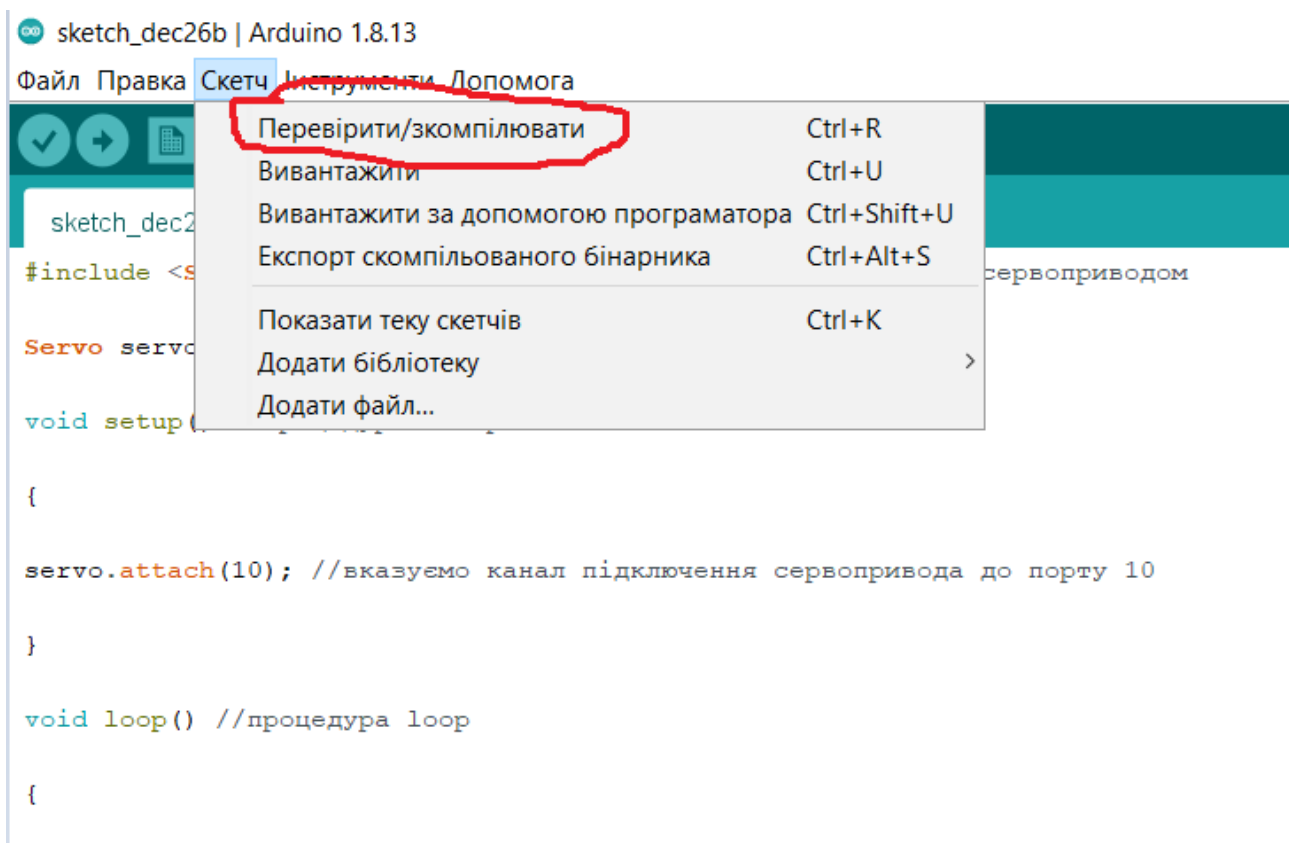


Рис.3.11 – Перевірка та компіляція програмного коду в Arduino IDE.

5. Завантажити розроблений програмний код до плати керування з середовища Arduino IDE (рис.3.12)

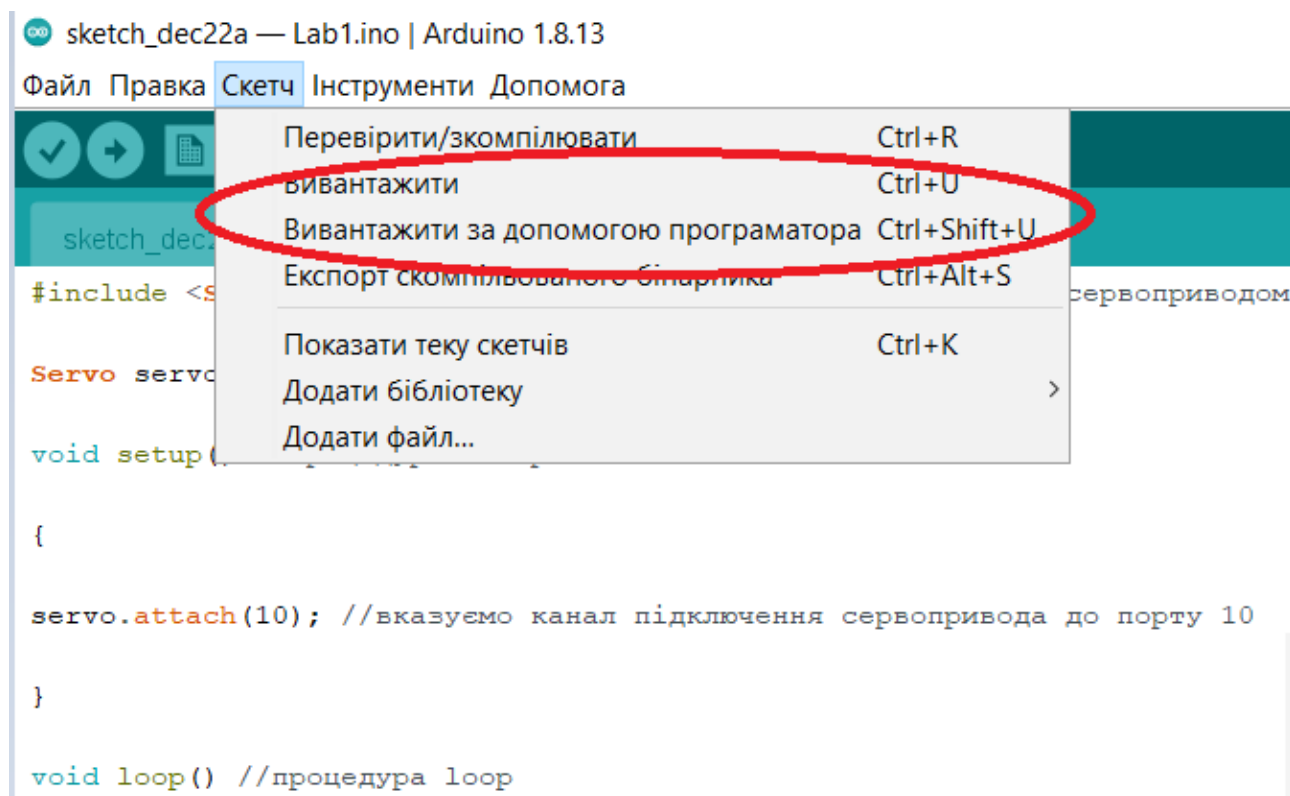


Рис.3.12 – Завантаження програмного коду з Arduino IDE до плати керування.

6. Після завантаження програмного коду, перевірити правильність виконання поставленої задачі на кут повороту вихідного валу.

7. За допомогою функції реєстрації сигналу керування «Послідовний плотер» (рис.3.13) сформувати графіки сигналів керування сервоприводом.

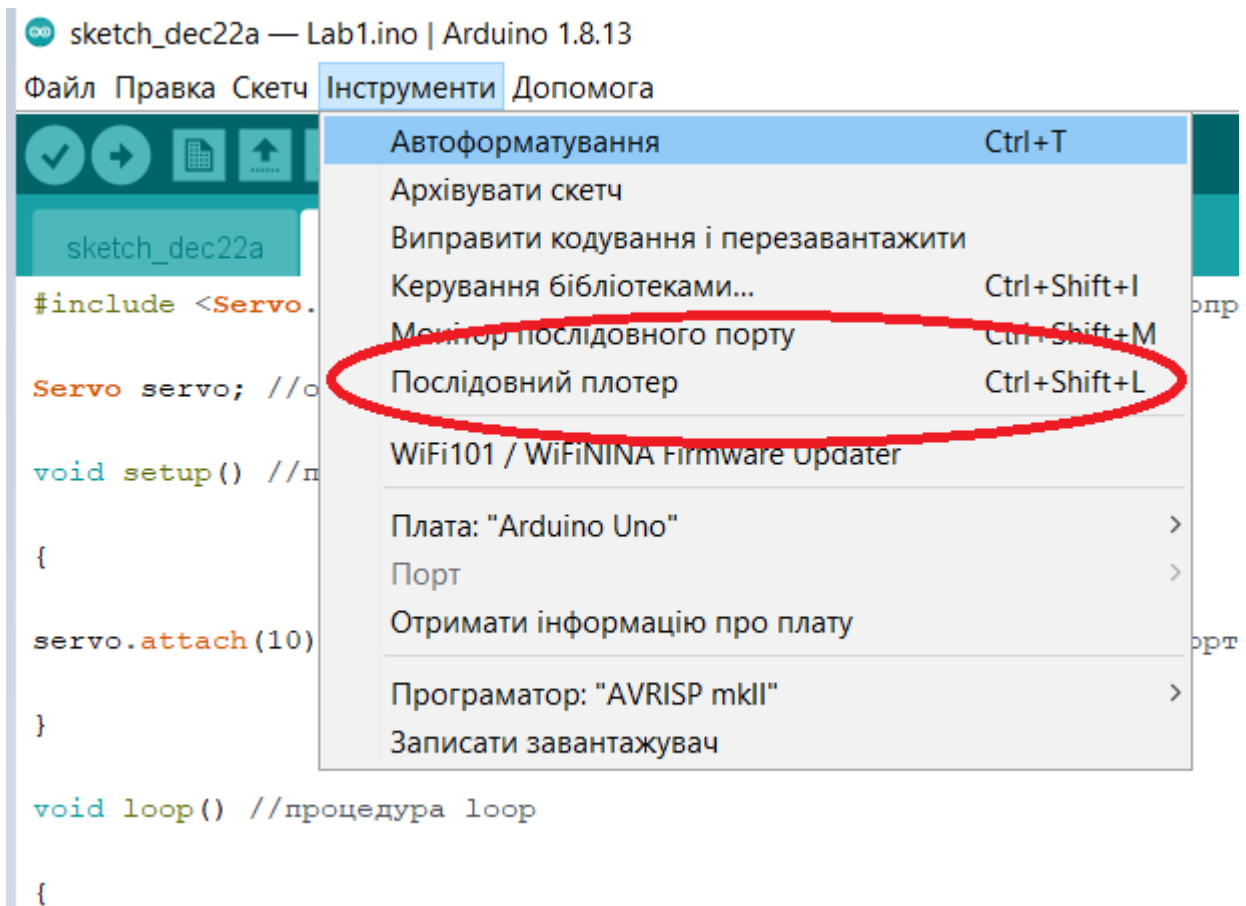


Рис.3.13 – Реєстрація вихідного сигналу в Arduino IDE за допомогою послідовного плотеру.

8. Підготувати звіт з лабораторної роботи.

#### Зміст звіту з лабораторної роботи.

1. Мета та завдання до лабораторної роботи.
2. Розроблена схема керування сервоприводом.
3. Програмний код (скетч) з коментарями до кожного рядка.
4. Графіки вихідних сигналів з коментарями.
5. Висновки до лабораторної роботи.
6. Результати завдань до самостійного виконання.



### Завдання для самостійного виконання.

1. Розробити програму керування та дослідити паралельну роботу сервоприводів згідно схеми (рис.3.14)

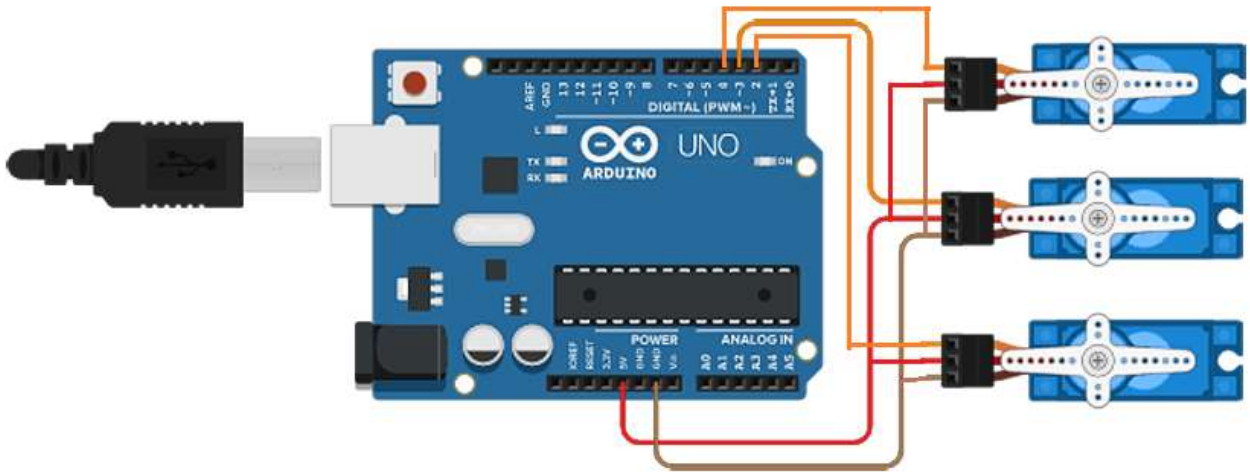


Рис.2.14 – Схема підключення сервоприводів до плати керування.

2. Розробити програму керування та дослідити роботу взаємопов'язаних сервоприводів згідно конструкції (рис.3.15)

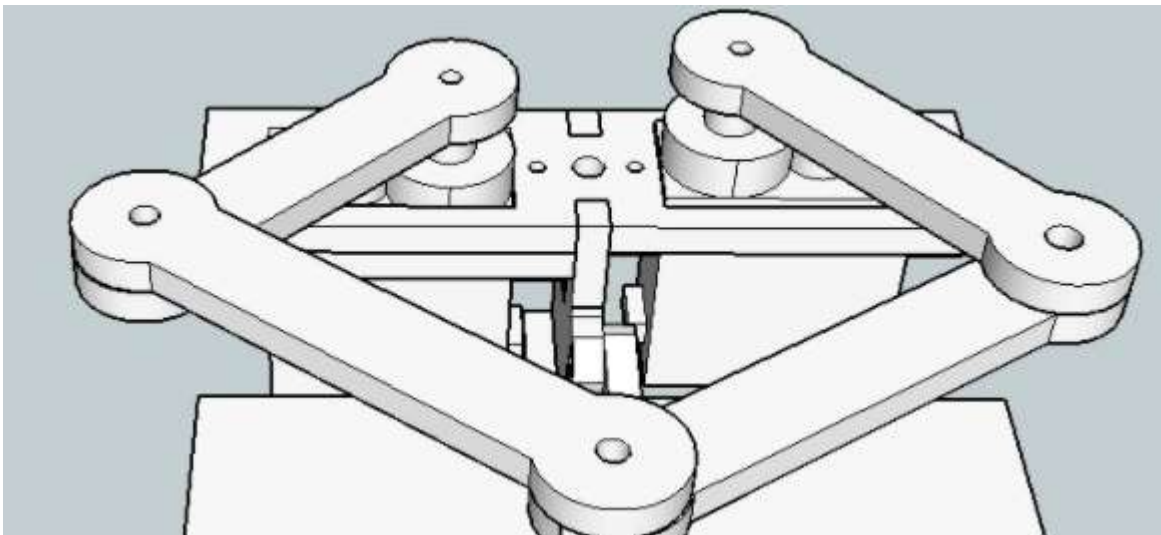


Рис.2.15 – Взаємопов'язані сервоприводи (2-Д робот).

**Лабораторна робота №3.** Дослідження роботи сервоприводів з редуктором MG996R з системою керування на базі Arduino.

**Мета роботи** – дослідити роботу сервоприводу типу MG996R з використанням системи керування на базі платформи Arduino Uno.

**Постановка задачі** – виконати підключення основних елементів електромеханічної системи та розробити програмний код для системи керування.

**Порядок виконання роботи**

**1.** Вибрати необхідні елементи та з'єднати їх згідно схеми підключення (рис.3.16)

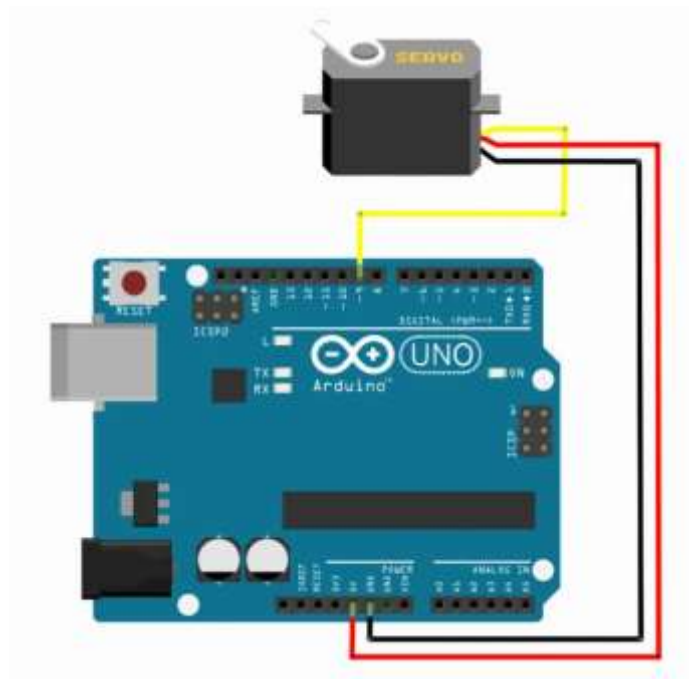


Рис.3.16 – Схема підключення сервоприводу до плати керування.

2. Розробити програмний код системи керування для забезпечення повороту валу сервопривода згідно діаграм (рис.3.17).

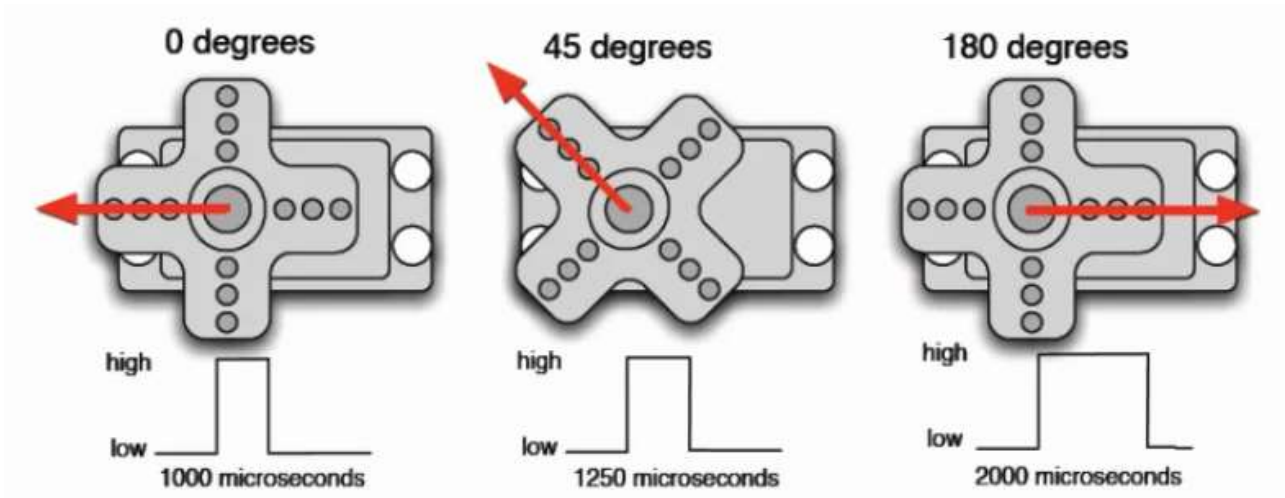


Рис.3.17 – Діаграма сигналів керування сервоприводом.

**Приклад програмного коду:**

```
#include <Servo.h>
Servo myservo; // Створюємо об'єкт керування типу серво
// до 12 таких об'єктів
int pos = 0; // змінна для зберігання положення сервоприводу
void setup()
{
  myservo.attach(9); // вказуємо порт підключення сервоприводу 9
}
void loop()
{
  for(pos = 0; pos <= 180; pos += 1) // робота від 0 градусів до 180 градусів
  { // з кроком 1 градус
    myservo.write(pos); // команда сервоприводу перейти до позиції у змінній 'pos'
    delay(15); // затримка 15 мс, поки сервопривід досягне позиції
  }
  for(pos = 180; pos >= 0; pos -= 1) // робота від 180 градусів до 0 градусів
  {
    myservo.write(pos); // команда сервоприводу перейти до позиції у змінній 'pos'
    delay(15); // затримка 15 мс, поки сервопривід досягне позиції
  }
}
```

3. Підключити плату керування Arduino Uno до ПК, (рис.3.18).

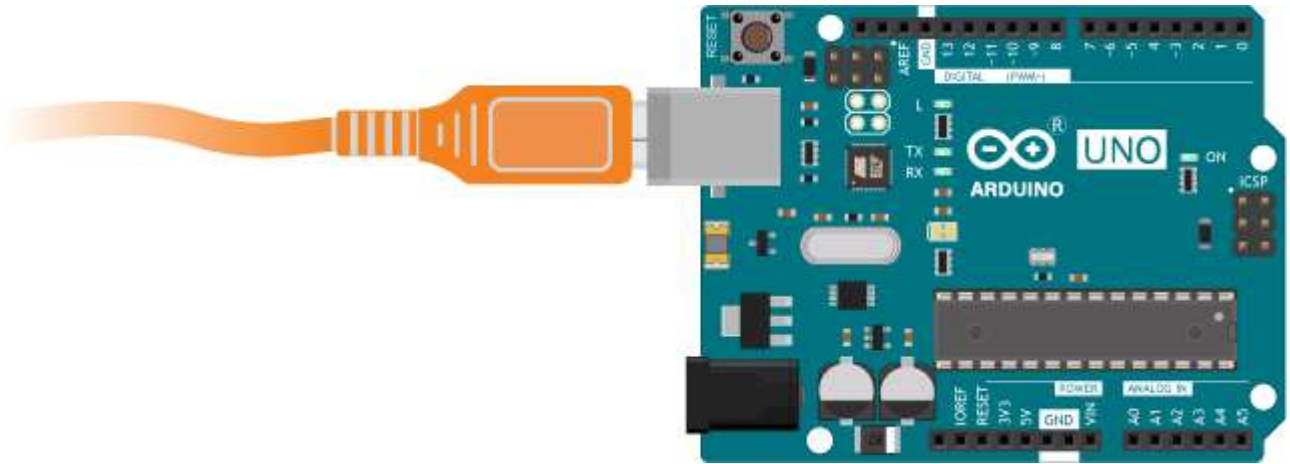


Рис.3.18 – Схема підключення плати керування Arduino Uno до ПК.

4. Провести попередню перевірку та компіляцію розробленого програмного коду у середовищі Arduino IDE (рис.3.19)

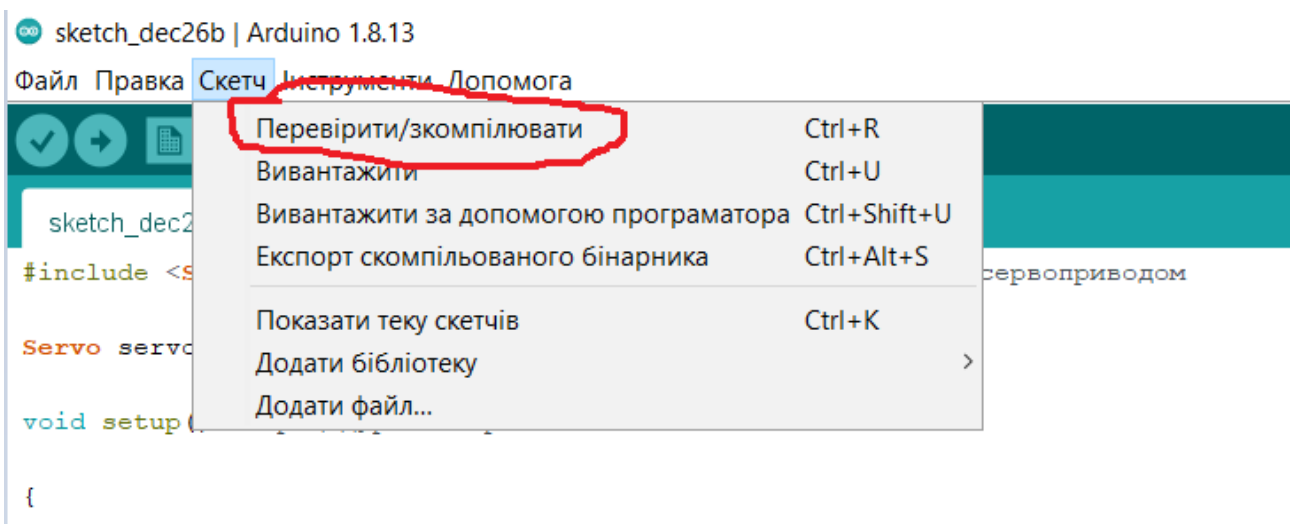


Рис.3.19 – Перевірка та компіляція програмного коду в Arduino IDE.

5. Завантажити розроблений програмний код до плати керування з середовища Arduino IDE (рис.3.20)

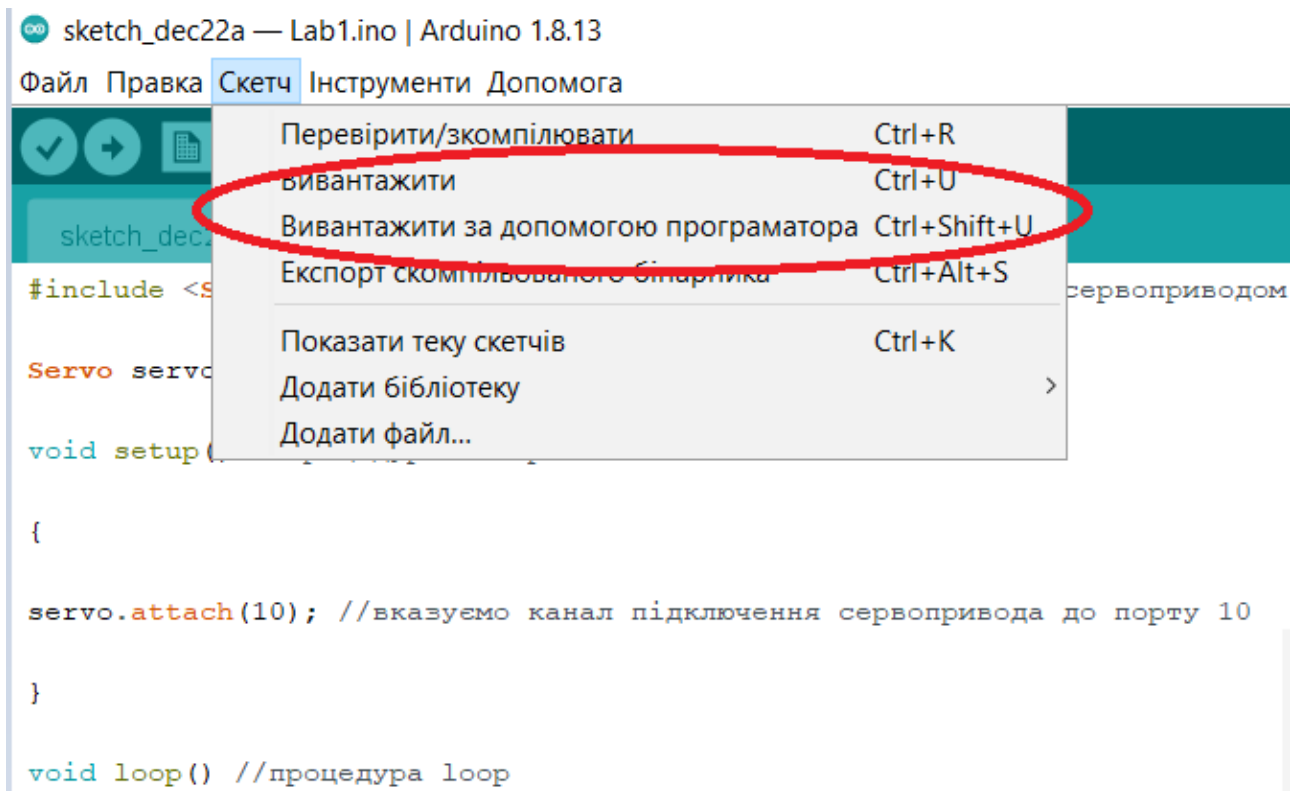


Рис.3.20 – Завантаження програмного коду з Arduino IDE до плати керування.

6. Після завантаження програмного коду, перевірити правильність виконання поставленої задачі на кут повороту вихідного валу.
7. За допомогою функції реєстрації сигналу керування «Послідовний плотер» (рис.3.21) сформувати графіки сигналів керування сервоприводом.

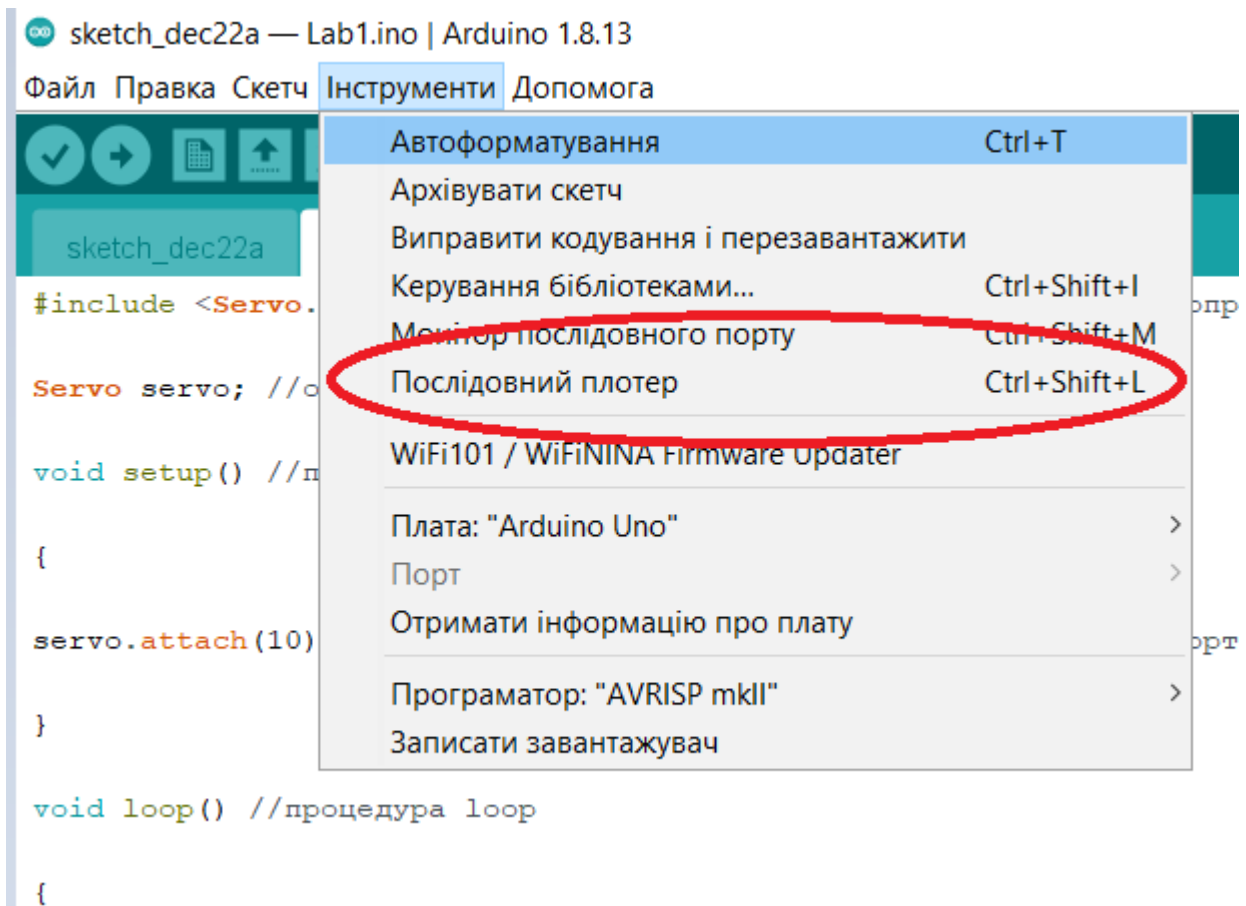


Рис.3.21 – Реєстрація вихідного сигналу в Arduino IDE за допомогою послідовного плотеру.

8. Підготувати звіт з лабораторної роботи.

#### Зміст звіту з лабораторної роботи.

1. Мета та завдання до лабораторної роботи.
2. Розроблена схема керування сервоприводом.
3. Програмний код (скетч) з коментарями до кожного рядка.
4. Графіки вихідних сигналів з коментарями.
5. Висновки до лабораторної роботи.
6. Результати завдань до самостійного виконання.

### Завдання для самостійного виконання.

1. Дослідити роботу сервопривода при наявності сигналу керування від зовнішнього потенціометра (рис.3.22)

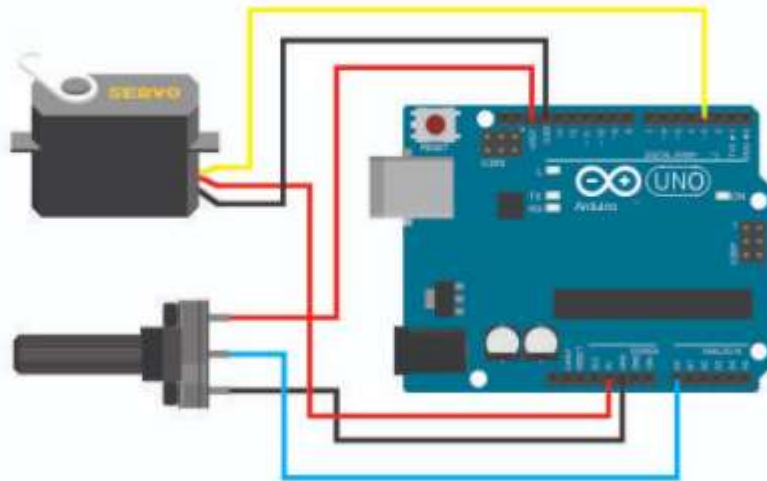


Рис.3.22 – Керування сервоприводом за допомогою зовнішнього потенціометра.

2. Розробити закон керування сервоприводом для забезпечення повороту вихідного валу в діапазоні від  $0^{\circ}$  до  $360^{\circ}$  з кроком  $10^{\circ}$  та затримкою часу між відділками в 1 сек. Дослідити роботу сервопривода при наявності такого сигналу керування.

## Лабораторна робота №4. Паралельна робота сервоприводів з редуктором MG996R з системою керування на базі Arduino.

**Мета роботи** – дослідити паралельну роботу сервоприводів типу MG996R з використанням системи керування на базі платформи Arduino Uno.

**Постановка задачі** – виконати підключення основних елементів електромеханічної системи та розробити програмний код для системи керування.

### Порядок виконання роботи

1. Вибрати необхідні елементи та з'єднати їх згідно схеми підключення (рис.3.23)

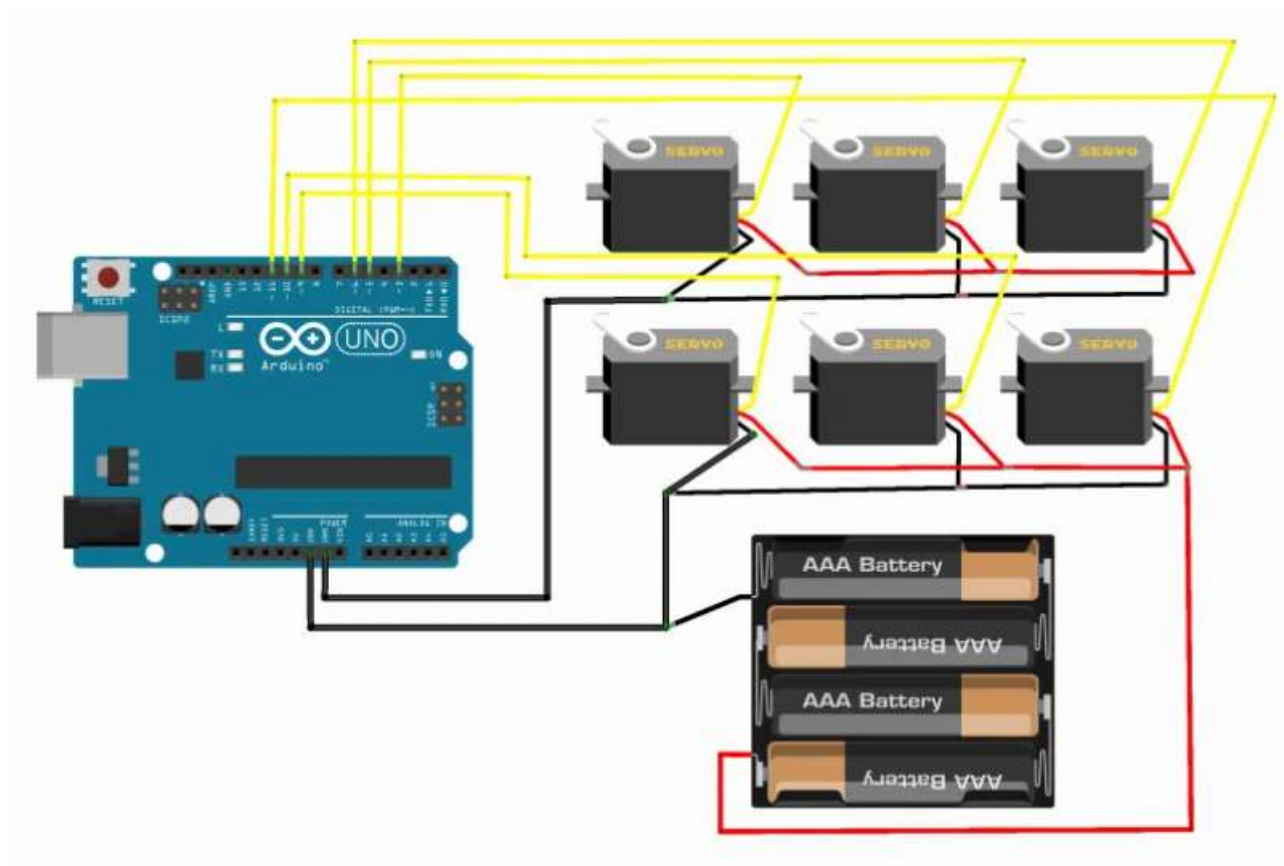


Рис.3.23 – Схема підключення сервоприводів до плати керування.

2. Розробити програмний код системи керування для забезпечення синхронного повороту валів сервоприводів згідно діаграм (рис.3.24).



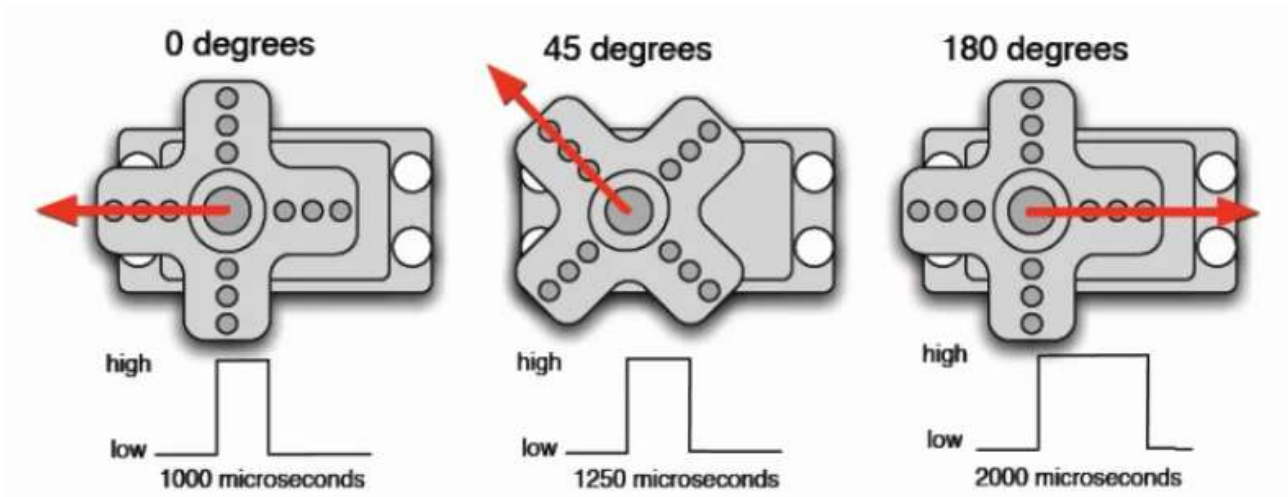


Рис.3.24 – Діаграма сигналів керування сервоприводами.

### Приклад програмного коду:

```
#include <Servo.h>
String readString;
Servo servo1, servo2, servo3, servo4, servo5, servo6; // Створюємо об'єкти керування
типу серво
void setup() {
  Serial.begin(9600);
  servo1.attach(3);
  servo2.attach(5);
  servo3.attach(6);
  servo4.attach(9);
  servo5.attach(10);
  servo6.attach(11);
}
void loop() {
  if (Serial.available()) {
    char c = Serial.read(); //отримує один байт із послідовного буфера
    if (c == ',') {
      if (readString.length() >1) {
        Serial.println(readString); //друкує рядок до послідовного порту
        int n = readString.toInt(); //перетворення readString into y число
        // auto select appropriate value, copied from someone elses code.
        if(n >= 500)
        {
          Serial.print("writing Microseconds: ");
          Serial.println(n);
          if(readString.indexOf('1') >0) servo1.writeMicroseconds(n);
          if(readString.indexOf('2') >0) servo2.writeMicroseconds(n);
          if(readString.indexOf('3') >0) servo3.writeMicroseconds(n);
          if(readString.indexOf('4') >0) servo4.writeMicroseconds(n);
          if(readString.indexOf('5') >0) servo5.writeMicroseconds(n);
          if(readString.indexOf('6') >0) servo6.writeMicroseconds(n);
        }
      }
      else
      {
        Serial.print("writing Angle: ");
        Serial.println(n);
      }
    }
  }
}
```

```

if(readString.indexOf('1') >0) servo1.writeMicroseconds(n);
if(readString.indexOf('2') >0) servo2.writeMicroseconds(n);
if(readString.indexOf('3') >0) servo3.writeMicroseconds(n);
if(readString.indexOf('4') >0) servo4.writeMicroseconds(n);
if(readString.indexOf('5') >0) servo5.writeMicroseconds(n);
if(readString.indexOf('6') >0) servo6.writeMicroseconds(n);
}
readString=""; //очищає змінну для нового введення
}
}
else {
readString += c; //робить рядок readString
}
}
}

```

3. Підключити плату керування Arduino Uno до ПК, (рис.3.25).

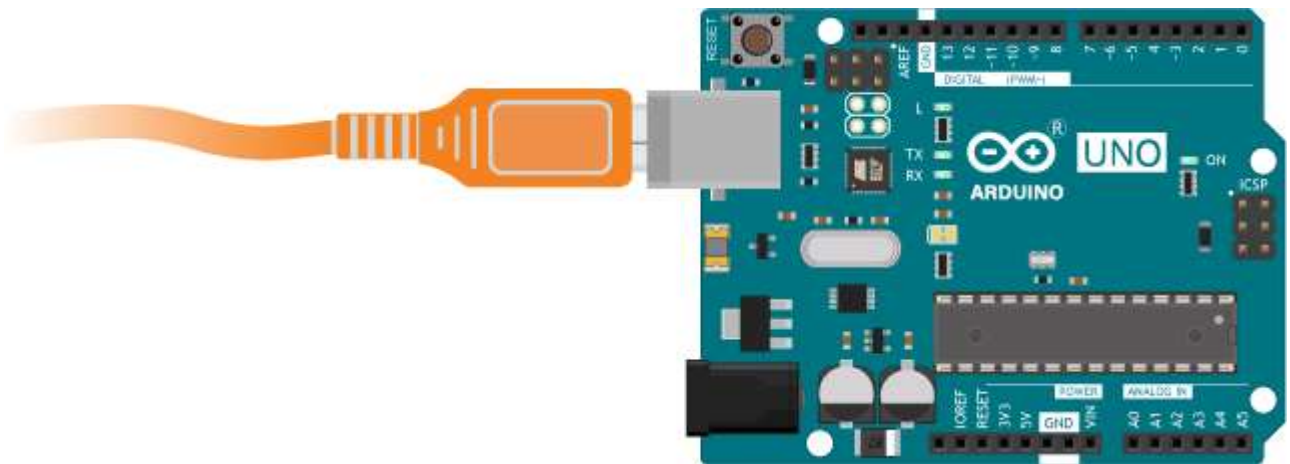


Рис.3.25 – Схема підключення плати керування Arduino Uno до ПК.

4. Провести попередню перевірку та компіляцію розробленого програмного коду у середовищі Arduino IDE (рис.3.26)

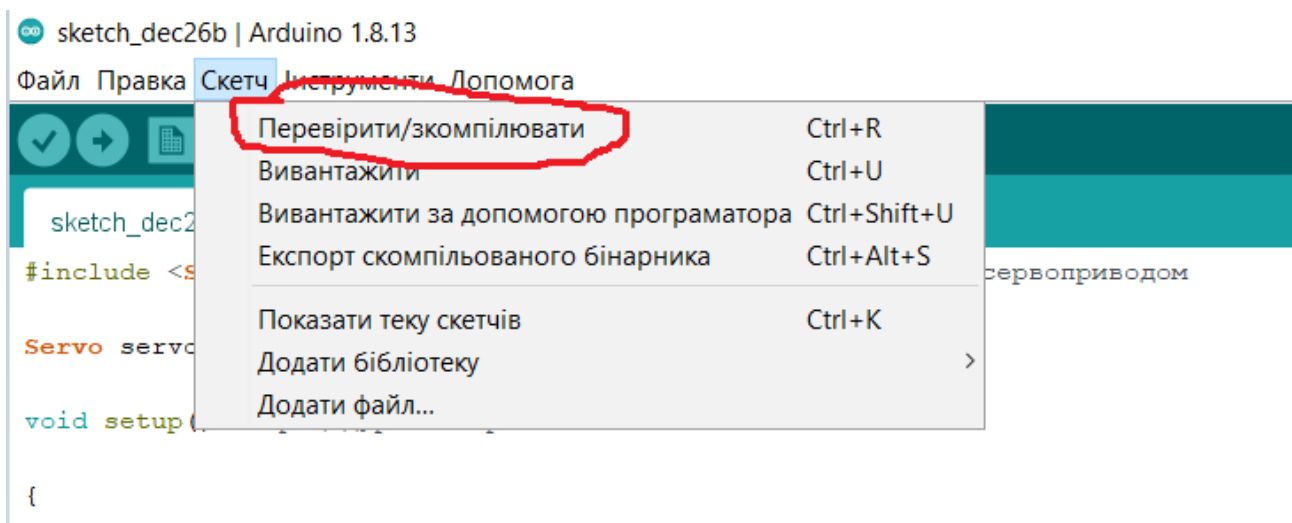


Рис.3.26 – Перевірка та компіляція програмного коду в Arduino IDE.

5. Завантажити розроблений програмний код до плати керування з середовища Arduino IDE (рис.3.27)

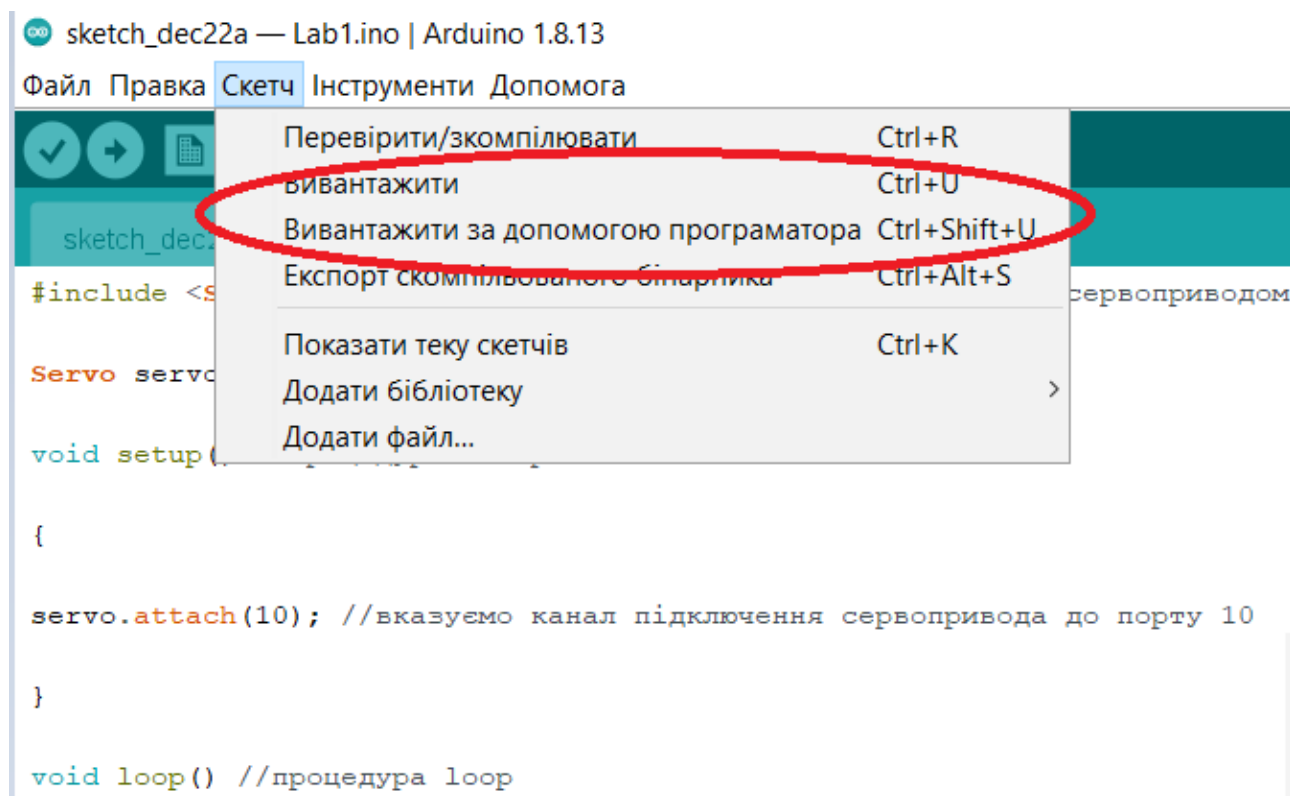


Рис.3.27 – Завантаження програмного коду з Arduino IDE до плати керування.

6. Після завантаження програмного коду, перевірити правильність виконання поставленої задачі на кут повороту вихідного валу.

7. За допомогою функції реєстрації сигналу керування «Послідовний плотер» (рис.3.28) сформувати графіки сигналів керування сервоприводом.

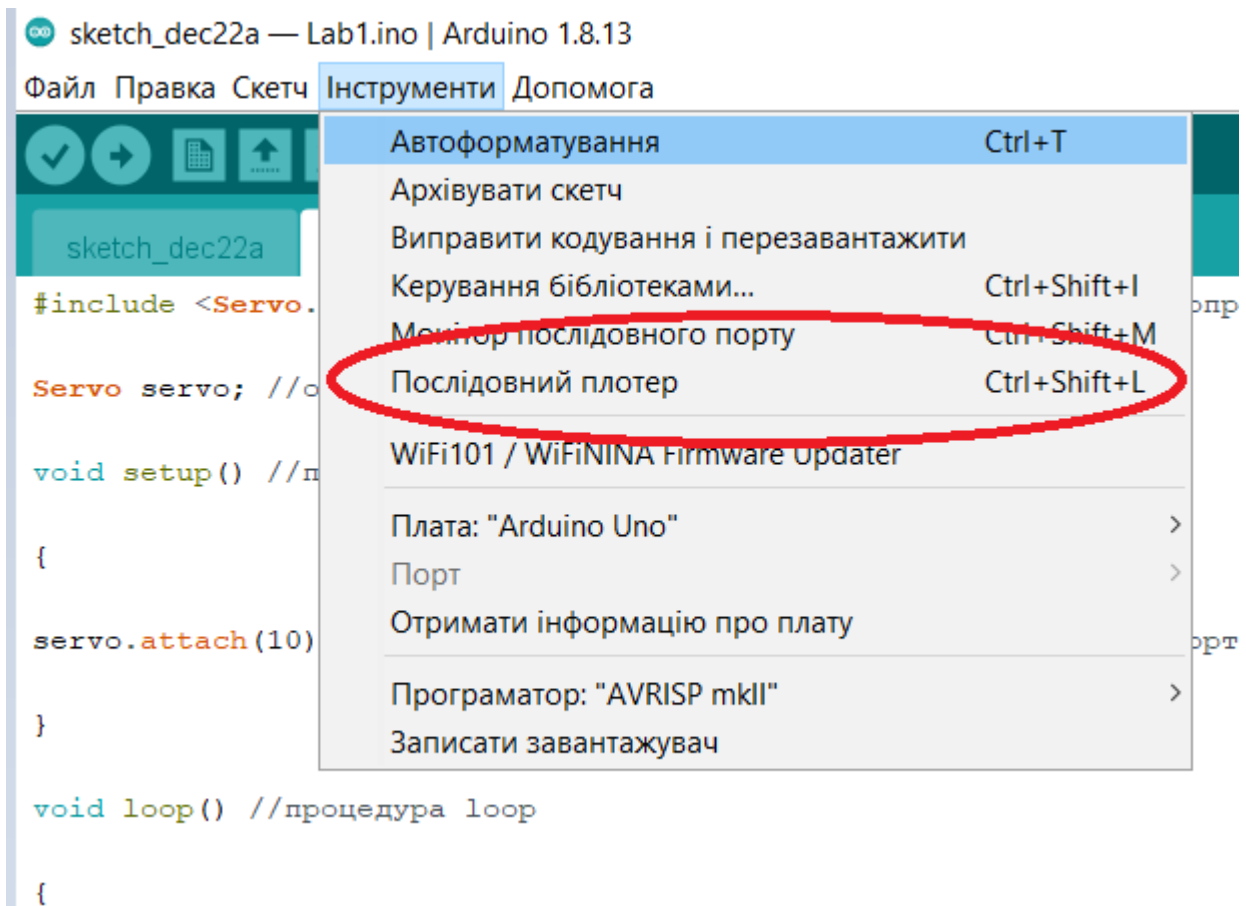


Рис.3.28 – Реєстрація вихідного сигналу в Arduino IDE за допомогою послідовного плотеру.

8. Підготувати звіт з лабораторної роботи.

#### **Зміст звіту з лабораторної роботи.**

1. Мета та завдання до лабораторної роботи.
2. Розроблена схема керування сервоприводом.
3. Програмний код (скетч) з коментарями до кожного рядка.
4. Графіки вихідних сигналів з коментарями.
5. Висновки до лабораторної роботи.
6. Результати завдань до самостійного виконання.

### Завдання для самостійного виконання.

1. Дослідити паралельну роботу сервоприводів при наявності сигналу керування від зовнішніх потенціометрів (рис.3.29)

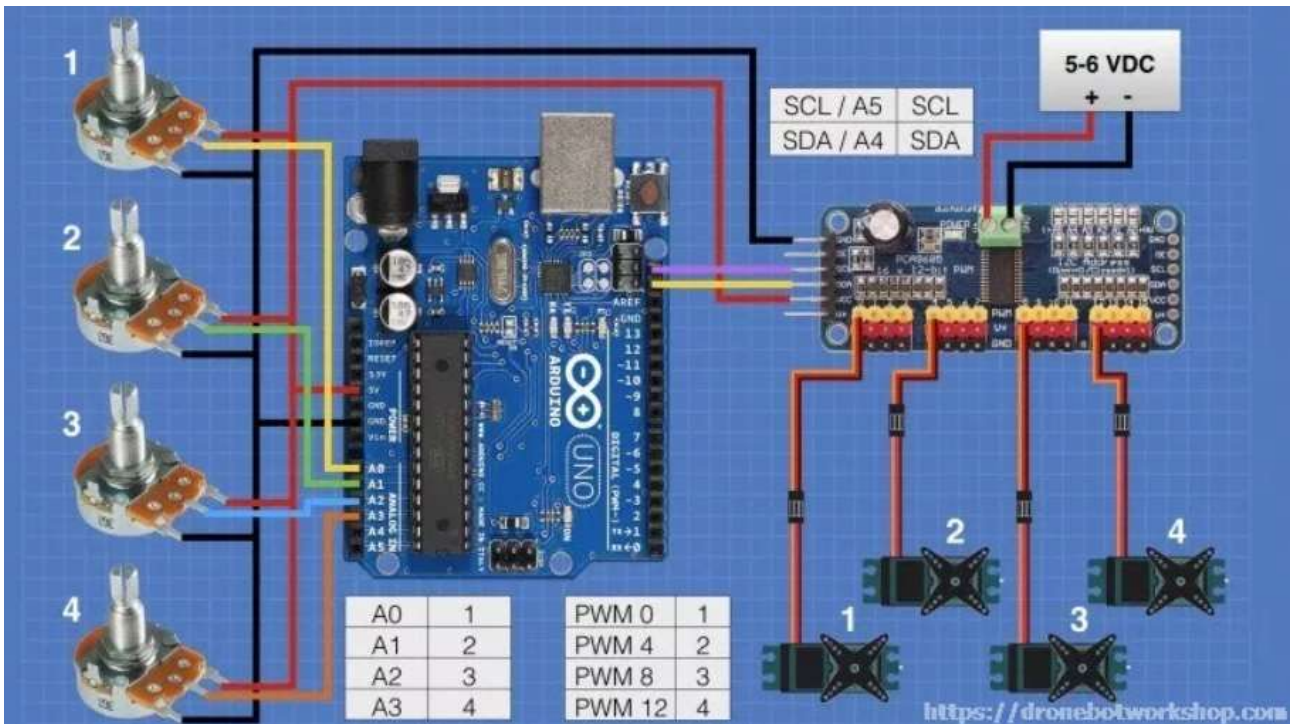


Рис.3.29 – Керування сервоприводами за допомогою зовнішніх потенціометрів.

**Лабораторна робота №5.** Дослідження роботи крокових двигунів STEP MOTOR 28BYJ-48 з системою керування на базі Arduino.

**Мета роботи** – дослідити роботу крокового двигуна типу STEP MOTOR 28BYJ-48 з використанням системи керування на базі платформи Arduino Uno.

**Постановка задачі** – виконати підключення основних елементів електромеханічної системи та розробити програмний код для системи керування.

**Порядок виконання роботи**

**1.** Вибрати необхідні елементи та з'єднати їх згідно схеми підключення (рис.3.30)

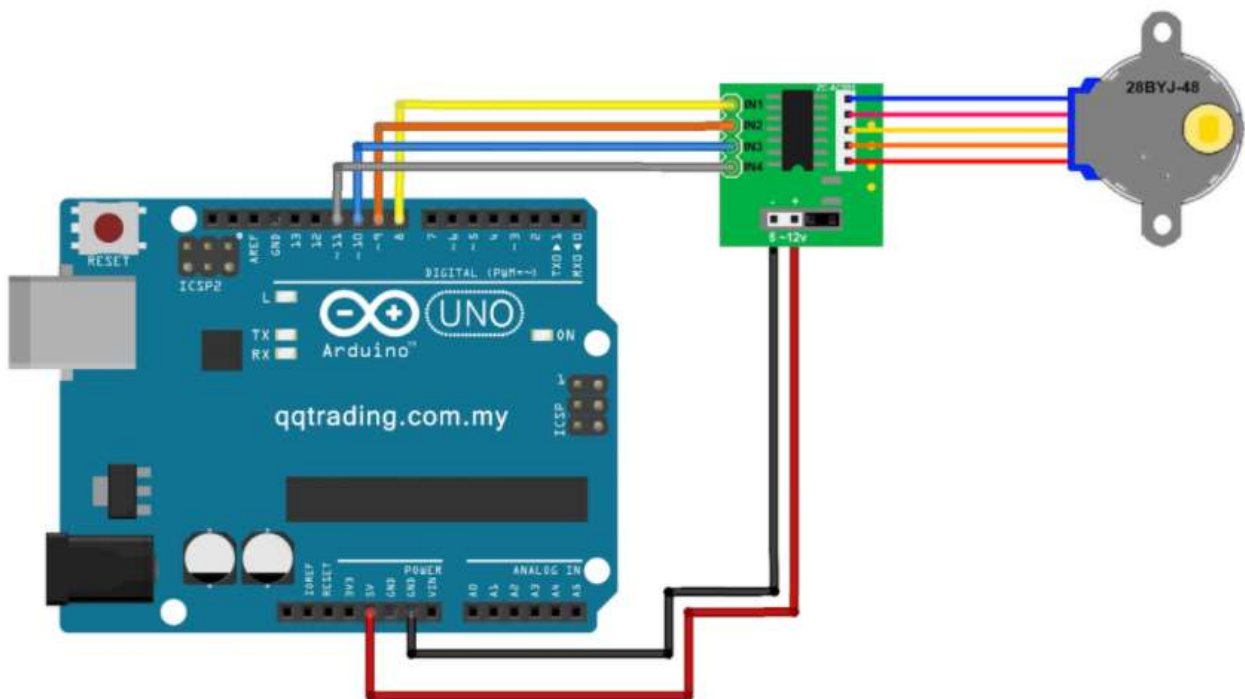


Рис.3.30 – Схема підключення крокового двигуна до плати керування.



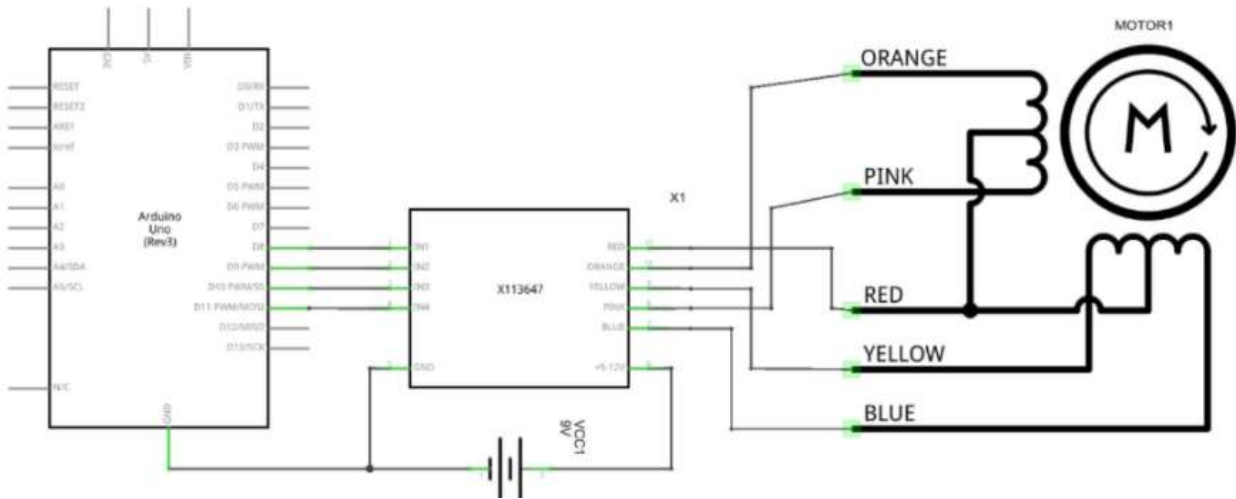
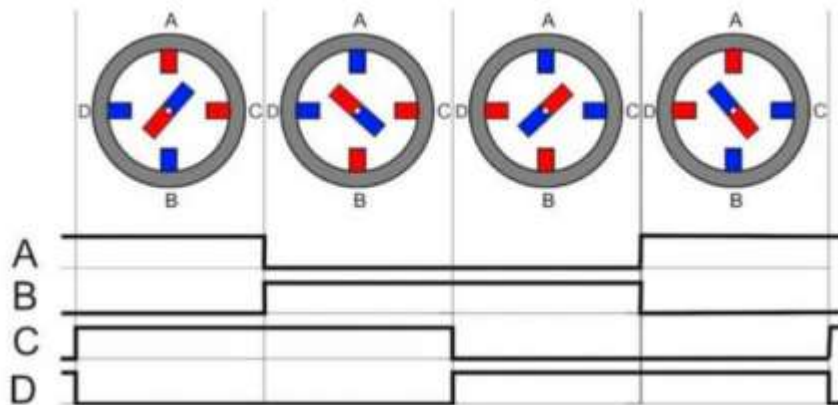


Рис.3.31 – Принципова схема підключення обмоток крокового двигуна.

2. Розробити програмний код системи керування для забезпечення повно крокового режиму роботи згідно діаграм (рис.3.32).



Контакт	Такты			
	1	2	3	4
A - orange	1	0	0	1
B - yellow	0	1	1	0
C - pink	1	1	0	0
D - blue	0	0	1	1

Рис.3.32 – Діаграма роботи крокового двигуна у повнокроковому режимі.

**Приклад програмного коду:**

```
#include <Stepper.h>
const int stepsPerRevolution = 200;
Stepper myStepper(stepsPerRevolution, 8,9,10,11); //підключення до портів 8...11 Ардуіно
void setup() {
myStepper.setSpeed(60); //завдання швидкості обертання ротора
Serial.begin(9600);
}
```

```

void loop() {
//функція очікує команди на рух мотора
Serial.println("Move right"); //за годинниковою стрілкою
myStepper.step(stepsPerRevolution);
delay(1000);
Serial.println("Move left"); //проти годин-вої стрілки
myStepper.step(-stepsPerRevolution);
delay(1000);
}

```

3. Підключити плату керування Arduino Uno до ПК, (рис.3.33).

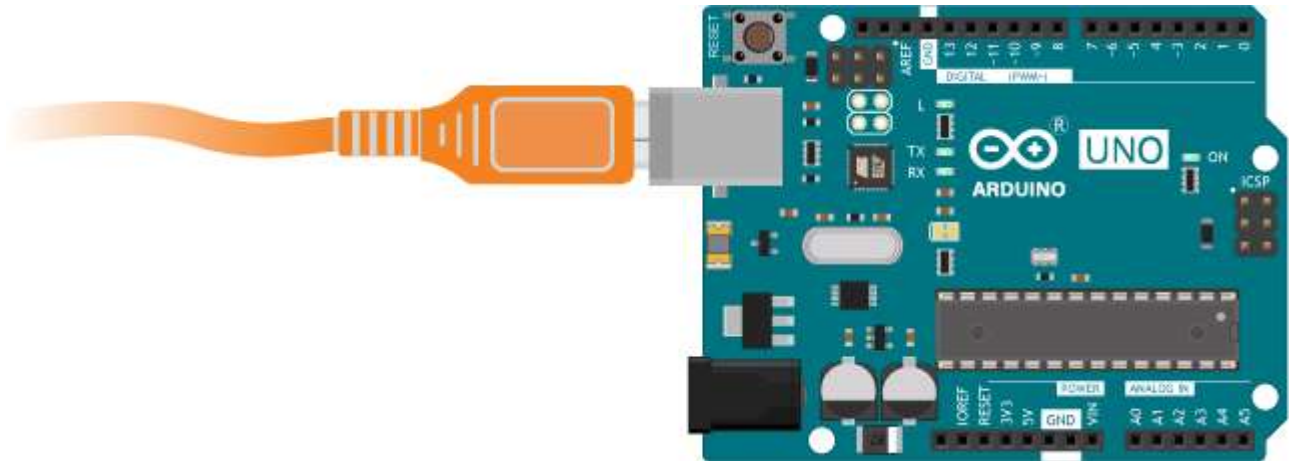


Рис.3.33 – Схема підключення плати керування Arduino Uno до ПК.

4. Провести попередню перевірку та компіляцію розробленого програмного коду у середовищі Arduino IDE (рис.3.34)

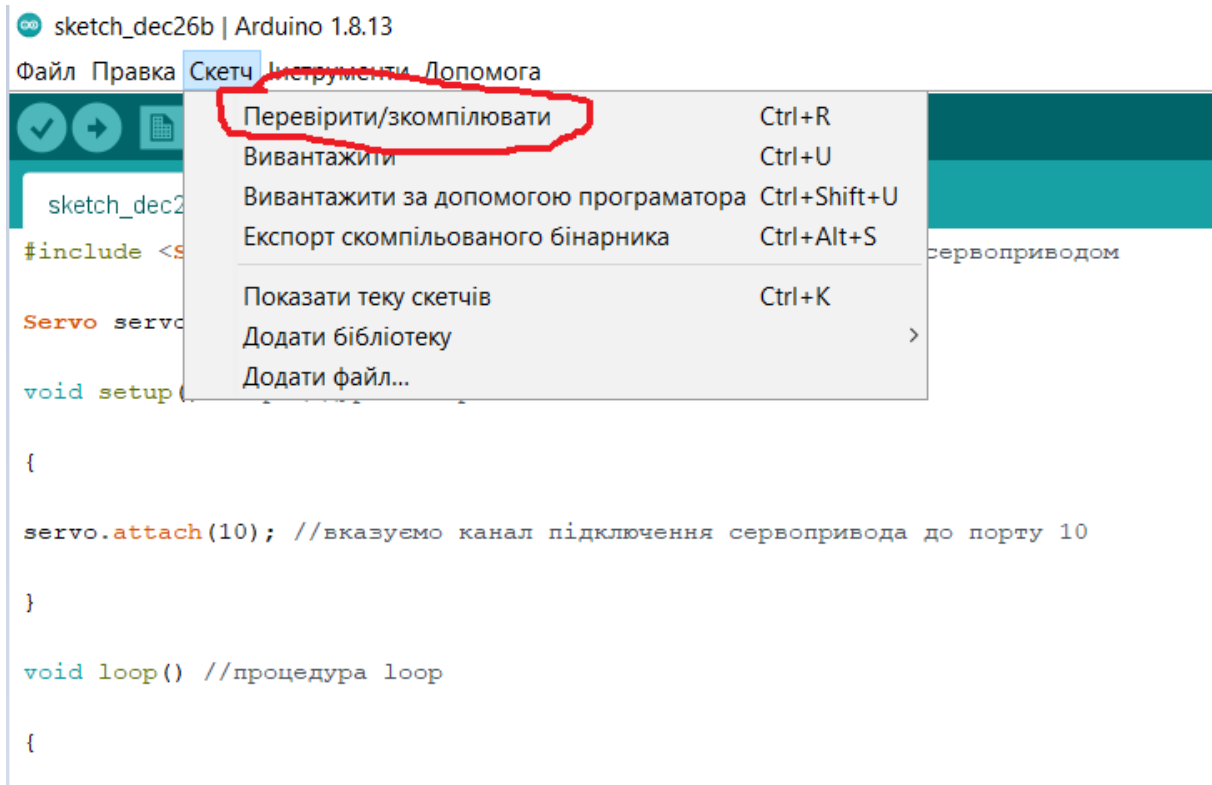


Рис.3.34 – Перевірка та компіляція програмного коду в Arduino IDE.



5. Завантажити розроблений програмний код до плати керування з середовища Arduino IDE (рис.3.35)

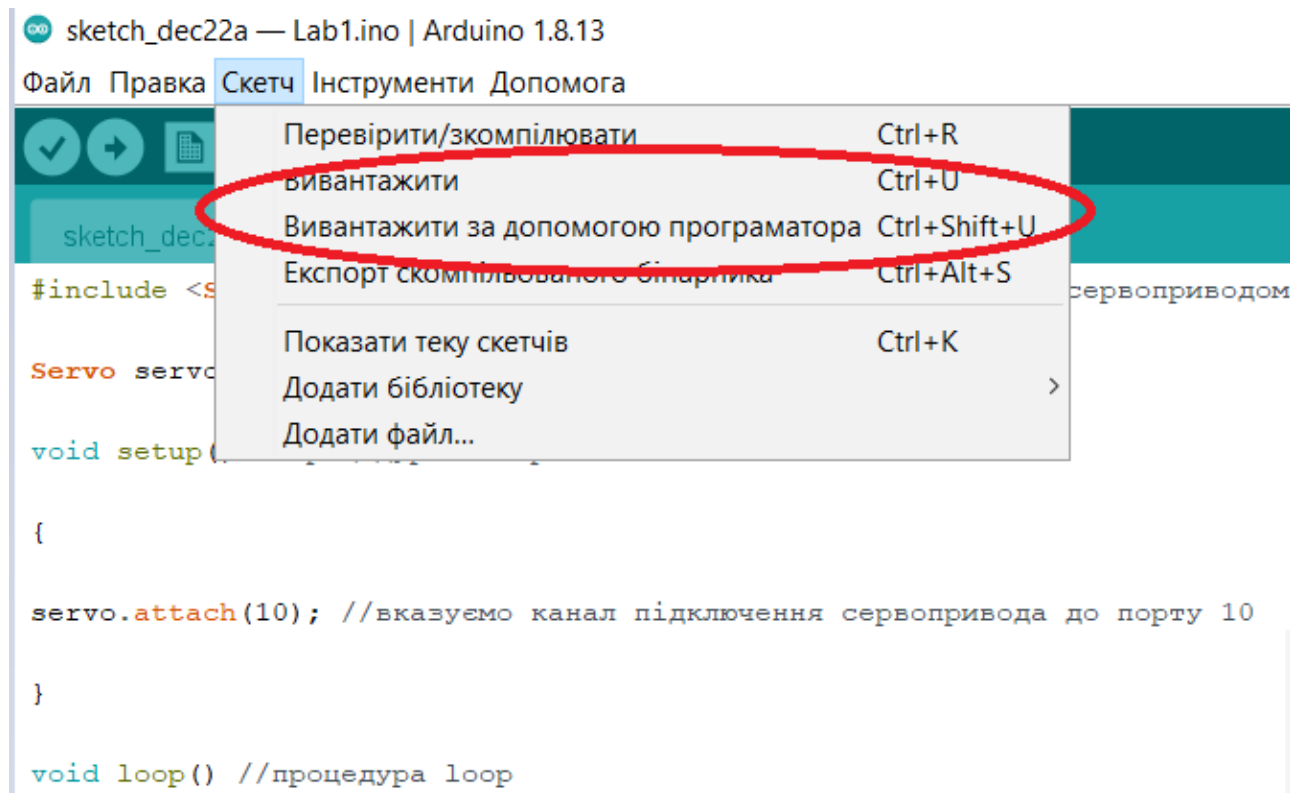


Рис.3.35 – Завантаження програмного коду з Arduino IDE до плати керування.

6. Після завантаження програмного коду, перевірити правильність виконання поставленої задачі на кут повороту вихідного валу.

7. За допомогою функції реєстрації сигналу керування «Послідовний плотер» (рис.3.36) сформувати графіки сигналів керування сервоприводом.

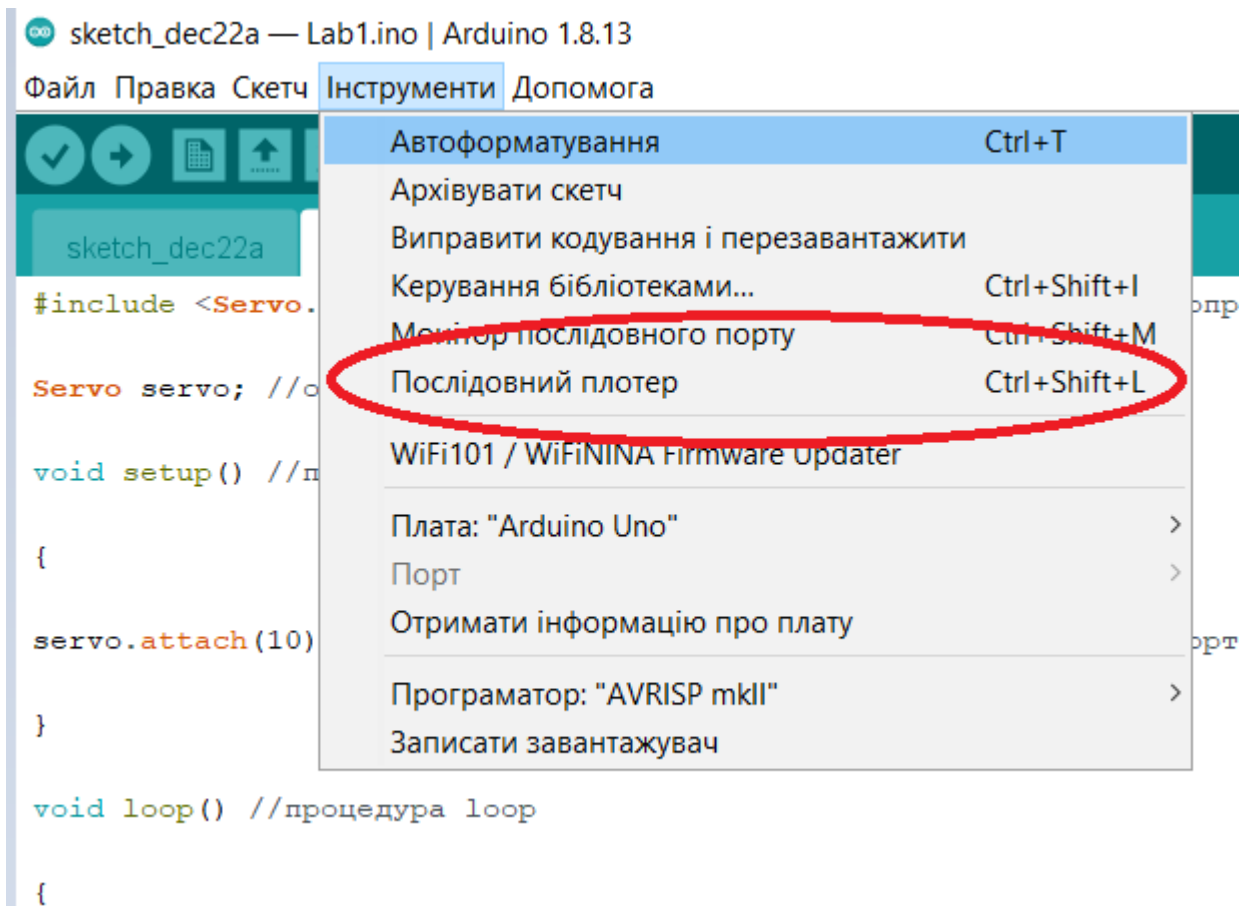


Рис.3.36 – Реєстрація вихідного сигналу в Arduino IDE за допомогою послідовного плотеру.

8. Підготувати звіт з лабораторної роботи.

#### Зміст звіту з лабораторної роботи.

1. Мета та завдання до лабораторної роботи.
2. Розроблена схема керування сервоприводом.
3. Програмний код (скетч) з коментарями до кожного рядка.
4. Графіки вихідних сигналів з коментарями.
5. Висновки до лабораторної роботи.
6. Результати завдань до самостійного виконання.

### Завдання для самостійного виконання.

1. Дослідити роботу крокового двигуна в напівкроковому режимі роботи згідно (рис.3.37)

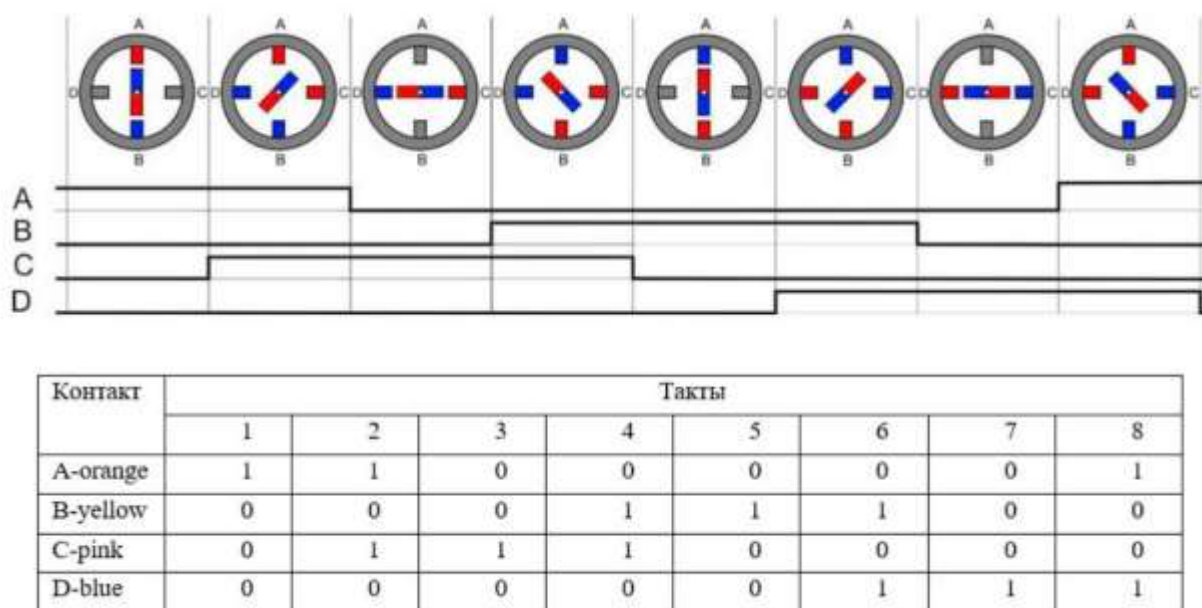


Рис.3.37 – Діаграма роботи крокового двигуна у напівкроковому режимі.

2. Розробити закон керування кроковим двигуном згідно схеми (рис.5.38) для забезпечення прямого, зворотного рухів та зупинки у відповідності до натискання конкретної кнопки. Дослідити роботу крокового двигуна при наявності різних сигналів керування.

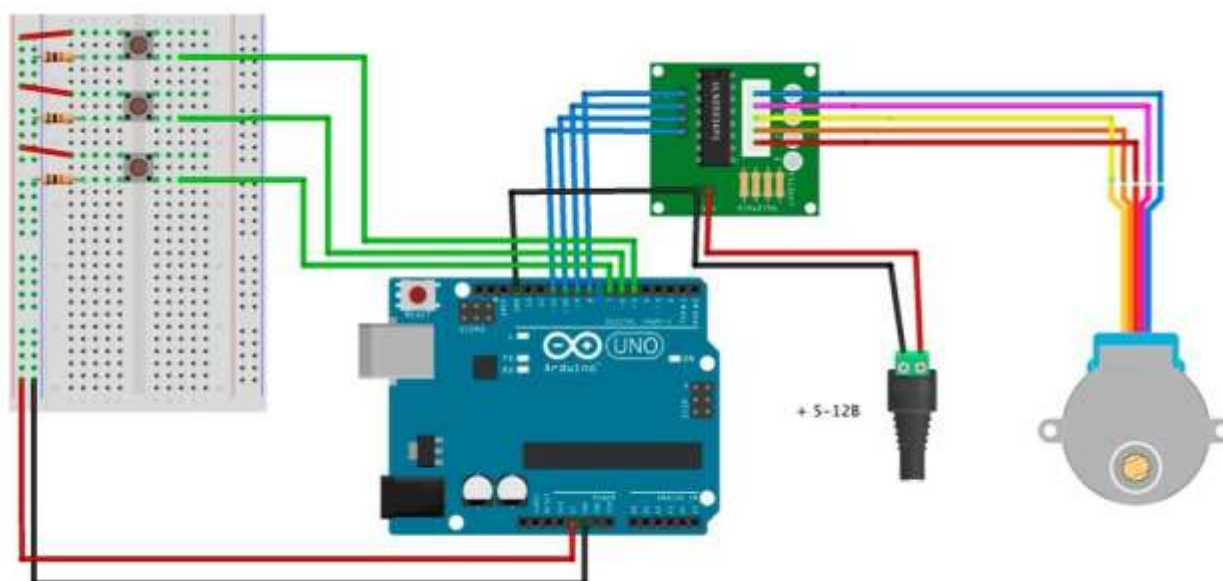


Рис.3.38 – Керування режимами роботи крокового двигуна за допомогою зовнішніх імпульсів.

## **Лабораторна робота №6.** Знайомство з програмним забезпеченням для програмування сервоприводів DELTA..

**Мета роботи** – ознайомитись з можливостями програмного забезпечення ASDA SOFT для програмування та налагодження сервоприводів компанії DELTA.

**Постановка задачі** – виконати підключення до лабораторного стенду лабораторії, провести дослідження згідно відеоінструкції.

### **Порядок виконання роботи**

1. Ознайомитись з відео інструкціями до програмного забезпечення.  
(<https://www.youtube.com/watch?v=oTVQ3D50GIE>,  
[https://www.youtube.com/watch?v=uHzGGNq\\_nm8](https://www.youtube.com/watch?v=uHzGGNq_nm8) )
2. Ознайомитись з інструкціями до експлуатації сервоприводу ASDA A2.
3. Провести підключення лабораторного стенду до мережі живлення та ПК.
4. Провести дослідження базових функцій користувача.
5. Підготувати звіт з лабораторної роботи.

## **Лабораторна робота №7.** Керування кроковими двигунами за допомогою лабораторного стенду віртуальної лабораторії Ройтлінгенського університету.

**Мета роботи** – дослідити паралельну роботу крокових двигунів з однією системою керування.

**Постановка задачі** – виконати підключення до лабораторного стенду віртуальної лабораторії, провести дослідження згідно онлайн інструкції.

### **Порядок виконання роботи**

1. Виконати підключення до лабораторного стенду віртуальної лабораторії шляхом переходу за посиланням:  
<https://vvl.reutlingen-university.de/homepage/de/index.html#demos/stepperMotor/main>
2. Провести дослідження згідно онлайн інструкції, візуально відслідковувати результати зміни вихідних даних.
3. Підготувати звіт з проведених досліджень.

**Лабораторна робота №8.** Дослідження можливостей системи позиціонування для утримання кулі на колесі за допомогою лабораторного стенду віртуальної лабораторії Ройтлінгенського університету.

**Мета роботи** – дослідити режими роботи системи керування сервомотором, яка використовує технологію управління, для забезпечення балансування кулі на колесі.

**Постановка задачі** – виконати підключення до лабораторного стенду віртуальної лабораторії, провести дослідження згідно онлайн інструкції.

### **Порядок виконання роботи**

1. Виконати підключення до лабораторного стенду віртуальної лабораторії шляхом переходу за посиланням:  
<https://vvl.reutlingen-university.de/homepage/de/index.html#demos/ballOnWheel/main>
2. Провести дослідження згідно онлайн інструкції, візуально відслідковувати результати зміни вихідних даних.
3. Підготувати звіт з проведених досліджень.

### **4. ОЦІНЮВАННЯ ЗВІТУ**

Здійснюється експертним методом з використанням критеріїв, регламентованих Положенням про оцінювання результатів навчання здобувачів вищої освіти НТУ «ДП» [2].

### **5. ЛІТЕРАТУРА**

1. Салов В.О. Макет методичних рекомендацій до виконання кваліфікаційних робіт : мет. посіб. для наук.-пед. пр-ів. / В.О. Салов ; Нац. техн. ун-т «Дніпровська політехніка». – Д. : НТУ «ДП», 2019. – 37 с.
2. Положення про організацію атестації здобувачів вищої освіти НТУ «Дніпровська політехніка» / М-во освіти і науки України, Нац. техн. ун-т. – Д. : НТУ «ДП», 2018. – 44 с.
3. Положення про оцінювання результатів навчання здобувачів вищої освіти Національного технічного університету «Дніпровська політехніка» (із змінами та доповненнями від 18.09.2018 та 11.12.2018, затвердженими Вченою радою університету)
4. Положення про навчально-методичне забезпечення освітнього процесу Національного технічного університету «Дніпровська політехніка» / М-во освіти і науки України, Нац. техн. ун-т. – Д.: НТУ «ДП», 2019. – 23 с.

5. Положення про систему запобігання та виявлення плагіату у Національному технічному університеті «Дніпровська політехніка» (зі змінами та доповненнями, затвердженими Вченою радою НТУ «Дніпровська політехніка» від 26.03.2019)

6. Stepping Motors a guide to theory and practice. 4<sup>th</sup> edition. Published by the Institution of Engineering and Technology. London. 2007/ ISBN 978-0-85296-417-0.

7. Drives and Control for Industrial Automation. Tan Kok Kiong, Andi Sudjana Putra. Springer-Verlag London Limited 2011. ISSN 1430-9491 ISBN 978-1-84882-424-9 DOI 10.1007/978-1-84882-425-6

8. DC Servos Application and Design with Matlab. Stephen M. Tobin. © 2011 by Taylor and Francis Group, LLC CRC Press is an imprint of Taylor & Francis Group, an Informa business International Standard Book Number-13: 978-1-4200-8005-6 (Ebook-PDF)

9. Hesse, S. und Schnell, G.: Sensoren für die Prozess- und Fabrikautomation. ViewegVerlag, Vieweg, 2004

#### Інформаційні ресурси:

1. Сторінка з вікіпедії сервопривод:  
[https://en.wikipedia.org/wiki/Servo\\_drive](https://en.wikipedia.org/wiki/Servo_drive)
2. Сторінка з вікіпедії сервомотори  
<https://en.wikipedia.org/wiki/Servomotor>
3. Технічні інформація про серводвигун. <https://realpars.com/servo-motor/>
4. Технічні інформація для керування сервоприводом.  
<https://realpars.com/servo-motors-advantages/>
5. Технічні інформація крокового двигуна. <https://realpars.com/stepper-motors-advantages/>
6. Технічні інформація плати керування сервоприводом.  
<https://www.teachmicro.com/arduino-servo-motor-tutorial/>
7. Технічні інформація плати керування на базі мікроконтролера ATmega2560.  
<https://doc.arduino.ua/ru/hardware/Mega2560>
8. Сайт компанії DELTA. <https://www.delta-electronics.info/ASDAA2>
9. Сайт кафедри електропривода:  
<https://elprivod.nmu.org.ua/ua/student/disciplines/servo.php>

**Навчально-методичне видання**

С.С. Худолій

О.С. Бешта

О.О. Азюковський

О.В. Балахонцев

Р.О. Боровик

**МЕТОДИЧНІ РЕКОМЕНДАЦІЇ  
ДО ЛАБОРАТОРНИХ РОБІТ  
З ДИСЦИПЛІНИ  
«СЕРВОПРИВОД»**

спеціальності 141 «Електроенергетика, електротехніка та електромеханіка»

Підготовлено до виходу в світ  
у Національному технічному університеті  
«Дніпровська політехніка».

Свідоцтво про внесення до Державного реєстру ДК № 1842  
49050, м. Дніпро, просп. Д. Яворницького, 19