

Міністерство освіти й науки України  
Донбаська державна машинобудівна академія  
Кафедра автоматизації виробничих процесів

## **МЕТОДИЧНІ ВКАЗІВКИ**

**до комп'ютерного практикуму**

по дисципліні

# **ЦИФРОВІ СИСТЕМИ КЕРУВАННЯ Й ОБРОБКИ ІНФОРМАЦІЇ**

**Частина 1**

## **ПРОГРАМУВАННЯ Й КОНФІГУРУВАННЯ СИСТЕМ ЧПУ**

(для студентів спеціальності 151 «Автоматизоване  
керування технологічними процесами»)

Краматорськ 2017

## УДК 681.5

Методичні вказівки до комп'ютерного практикуму по дисципліні "Цифрові системи керування й обробки інформації". Частина 1: Програмування й конфігурування систем ЧПУ (для студентів спеціальності 151 "Автоматизоване керування технологічними процесами») / Уклад. О. О. Сердюк. - Краматорськ: ДДМА, 2017 – 89 с.

У методичних вказівках приводяться принципи створення програми керування автоматикою верстата мовою програмування S7-Graph. З використанням прикладів показані методики створення графів стану для елементарних функціональних одиниць устаткування, графа стану, що координує весь процес керування, а також групового графа, що підтримує діагностичні й алармові функції. Наведена методика конфігурування системи ЧПУ SINUMERIK із приводами SIMODRIVE 611. Викладена методика проектування приводної системи SINAMICS S120 у середовищі програми SIZER.

Укладач

О. О. Сердюк, доц.,

## ЗМІСТ

1	СТВОРЕННЯ ПЕРВИННИКІВ МОВОЮ S7-HIGRAPH	4
1.1	Принцип програмування мовою S7-Higraph	4
1.2	Приклад виділення графів станів у завданні керування	6
1.3	Послідовність створення графа станів в Higraph	10
1.4	Вимоги до звіту по роботі	18
2	РОЗРОБКА Й НАЛАГОДЖЕННЯ ПРОГРАМИ HIGRAPH	20
2.1	Створення групового графа	20
2.2	Компіляція первинників і створення блоків програми	26
2.3	Завантаження програми в контролер і її налагодження	29
2.4	Порядок виконання роботи й вимоги до звіту	31
3	КОНФІГУРУВАННЯ СИСТЕМИ ЧПУ SINUMERIK	32
3.1	Інтерфейс конфігуратора NCSD	32
3.2	Приклад конфігурування системи ЧПУ	38
4	ПРОЕКТУВАННЯ ПРИВІДНОЇ СИСТЕМИ ВЕРСТАТА SINAMICS У СЕРЕДОВИЩІ ПРОГРАМИ SIZER	43
4.1	Розробка компонування приводної системи верстата	43
4.2	Конфігурування привода головного руху	49
4.3	Конфігурування приводів подач	58
4.4	Конфігурування модуля живлення	64
4.4	Вибір системних компонентів	67
4.5	Індивідуальні завдання й вимоги до звіту по роботі	78
	ДОДАТОК А. Базові функції STL ( до роботи 2)	82
	ДОДАТОК Б. Варіанти завдань до роботи 3	87
	ДОДАТОК В. Варіанти завдань до роботи 4	88

# 1 СТВОРЕННЯ ПЕРВИННИКІВ МОВОЮ S7-HIGRAPH

*Ціль роботи: освоєння інтерфейсу й придбання навичок створення первинних файлів (графів станів) у редакторі S7-Higraph – інструментальному додатку програмної системи STEP 7.*

## 1.1 Принцип програмування мовою S7-Higraph

Графічна нотація широко використовується для опису поведінки автоматів, що здійснюють логічне керування встаткуванням. Алгоритми керування часто представляються блок-схемами, графами станів, перемикальними схемами, мережами Петрі й т.п. Мова програмування S7-Higraph дозволяє розширити функціональну область середовища програмування STEP 7 шляхом застосування графічного методу програмування на основі використання графів станів.

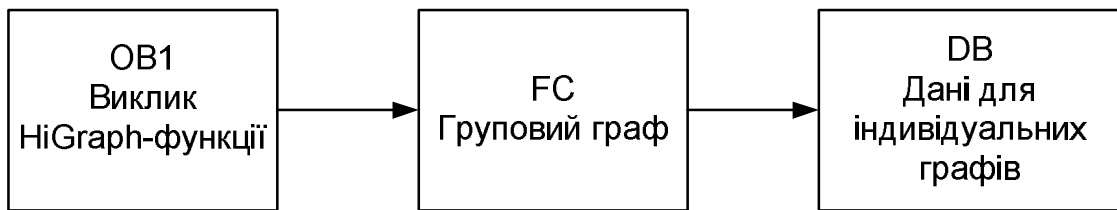
Для застосування цієї мови об'єкт автоматизації розділяється на функціональні матеріальні одиниці. Поведінка кожної функціональної одиниці описується графом станів (первинником). Для організації взаємодії графів станів створюється координуючий граф станів (граф-диспетчер). Усі процедури процесу створення графів станів здійснюються в редакторі S7-Higraph. Виконаний етап роботи з кожного графа зберігається в контейнері "Source files" (первинні файли) S7-програми без перевірки синтаксису.

Higraph-програма структурована в такий спосіб:

- У графах станів визначаються дії, які можуть бути зроблені при вході в стан, під час стану і при виході зі стану.
- Керування переходом (транзакцією) з одного стану в інший здійснюється розв'язною умовою із заданим рівнем пріоритету.
- Дії в станах і умови в транзакціях описуються мовою програмування STEP 7 STL.
- Для забезпечення взаємодії графів станів з координуючим графом станів програмуються повідомлення – вихідні в станах і вхідні в транзакціях.
- Графи станів разом з координуючим графом вставляються в груповий граф. У груповому графові призначаються поточні параметри всім змінним і повідомленням.
- Груповий граф компілюється зі створенням функції керування (FC) і блоку даних поточних значень параметрів (DB), які розміщуються в контейнері Blocks S7-програми.

Увесь алгоритм керування може бути задокументований у графічній і текстовій формі.

Функція Higraph FC повинна викликатися із циклічно працюючого блоку OB1. Структура готової програми показана на рисунку 1.1.



*Рисунок 1.1 – Структура Ніgraph-програми*

Редактор Ніgraph забезпечує наступні функції програмування:

- Програмування станів і транзакцій мовою STL.
- Програмування викликів FC-функції Ніgraph і логічних блоків STEP 7 (FC, SFC, FB, SFB), створених із застосуванням мов STL, LAD, FBD, а також SCL-команд.
- Програмування часу очікування завершення дії й контрольного часу перебування в стані.
- Тестування функціональних блоків з визначенням активного стану, попереднього стану й останньої транзакції, а також з виставою інформації щодо команд у станах і транзакціях.
- Виявлення помилок процесу, блокувань за часом і аварійних ситуацій з виводом інформації на пристрій зв'язку з оператором.

Графи станів і групові графи зберігаються в SIMATIC Manager у контейнері "Source files", а скопійовані групові графи у вигляді функції FC, блоку даних DB і додаткових блоків – у папці Blocks.

Редактор Ніgraph забезпечує наступні опції програмування:

- Вставку будь-якої кількості графів стану в груповий граф.
- Одночасне редагування декількох групових графів.
- Програмування умов у транзакціях.
- Програмування дій у станах, причому дії характеризуються подіями на вході (E), діями на виході (X) і циклічними діями (C и C-).
- Використання для програмування всього спектра STL команд. Перелік STL-команд наведений у додатку.
- Уведення контрольного часу й часу очікування у формі змінних або констант.
- Перемикання між символічною й абсолютною виставою адрес.

*Послідовність створення програми містить у собі наступні кроки:*

1. Створення проекту програми в Simatic Manager.
2. Вставка графа станів у програму.
3. Визначення сигналів, необхідних для керування.
4. Програмування станів.

5. Програмування транзакцій.
6. Програмування постійних інструкцій.
7. Програмування операційних режимів (автоматичний, ручний).
8. Створення координуючого графа.
9. Створення групового графа.
10. Установка послідовності виконання.
11. Призначення фактичних параметрів.
12. Програмування повідомлень.
13. Компіляція первинників і створення блоків програми.
14. Завантаження програми в контролер.
15. Налаштування програми в інтерактивному режимі.

У першій роботі практикуму, розрахованій на 4 години занять, необхідно виконати кроки 1-8, у другій роботі, такої ж тривалості, завершити створення програми (кроки 9-15).

## 1.2 Приклад виділення графів станів у завданні керування

Розглянемо приклад, наведений у Посібнику користувача «S7-Higraph V5.3 Programming State Graphs. Programming and Operating Manual».

У цьому прикладі створюється проста програма для свердлильного верстата, показано на рисунку 1.2.

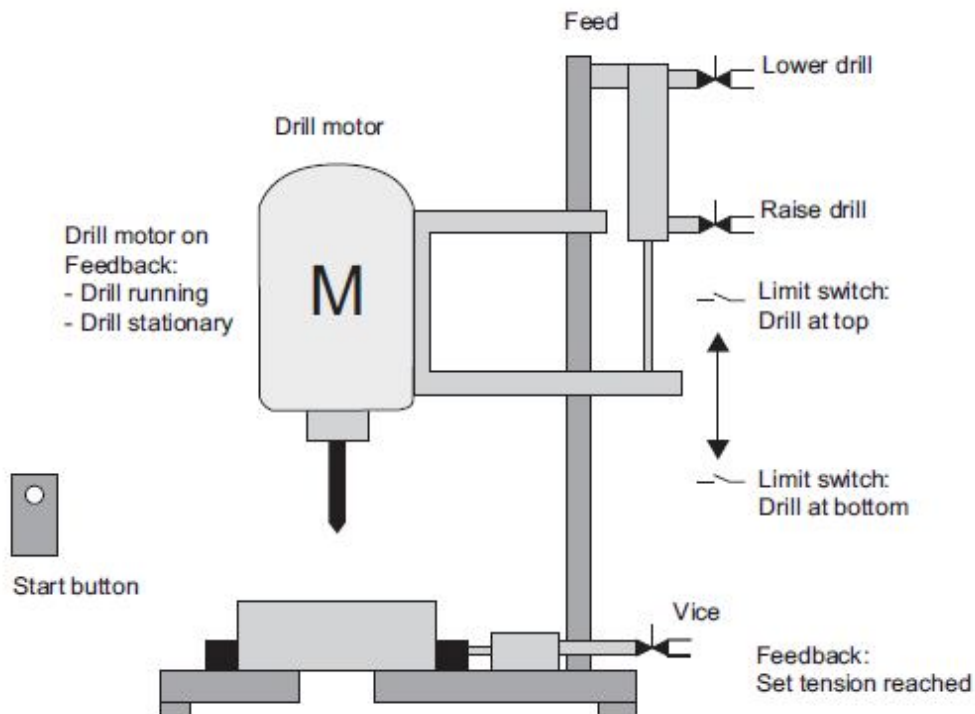


Рисунок 1.2 – Функціональні елементи свердлильного верстата

Свердильний верстат містить гідравлічний затискач заготовки (Vice, лещата), керований за допомогою золотника, електропривод обертання свердла (Drill motor) і гідравлічний привід подачі свердильної головки (Feed) із золотниками керування підйомом (Raise drill) і опусканням (Lower drill). Контроль процесу здійснюється кінцевими вимикачами переміщення свердильної головки (Limit switch), датчиком швидкості обертання вала двигуна (Motor Running) і тензодатчиком (Tension Reached), що сигналізують про досягнення необхідної сили затискання заготовки. Пуск процесу свердління здійснюється кнопкою Start button.

Початковий стан верстата визначений таким чином:

- Мотор привода свердла зупинений.
- Свердильна головка в крайньому верхньому положенні.
- Заготовка встановлена в затискнім пристосуванні, не затиснута.

На рисунку 1.3 показана функціональна діаграма процесу свердління, що складається із 8 станів.

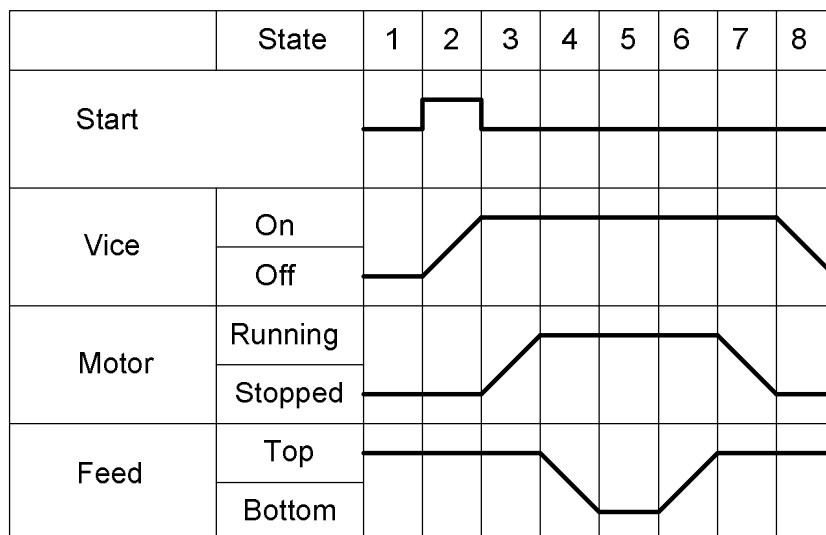


Рисунок 1.3 – Функціональна діаграма процесу свердління

Процес свердління в автоматичному режимі починається із включення верстата пусковою кнопкою Start button і складається з наступних операцій:

1. Затискання заготовки (поки тиск фіксації не досягнутий).
2. Пуск двигуна обертання свердла.
3. Подача свердла вниз, поки не досягнута крайня нижня точка.
4. Витримка в нижньому положенні.
5. Подача свердла нагору, поки не досягнута крайня верхня точка.
6. Зупинка двигуна обертання свердла.
7. Розтискання заготовки (заготовка видаляється оператором).

Виконавчі пристрої свердлильного верстата управляються через виходи модуля цифрового виводу з адресами Q 0.0 - Q 0.7. Вхідні сигнали подаються на модуль введення з адресами входів I 0.0 - I 0.7.

Призначення вхідних і вихідних сигналів наведено в таблиці 1.1.

Таблиця 1.1

Адреса		Опис
Символьна	Абсол.	
Drill_Motor_Running	I 0.0	Свердло обертається із заданою швидкістю
Drill_Motor_Stopped	I 0.1	Двигун привода свердла зупинений
Drill_at_Bottom	I 0.2	Свердлильна головка в нижньому положенні
Drill_at_Top	I 0.3	Свердлильна головка у верхньому положенні
Tension_Reached	I 0.4	Заготовка затиснута (тиск досягнутий)
Start_Button	I 0.7	Сигнал пускової кнопки
Drill_Motor_On	Q 0.0	Вмикати двигун привода свердла
Lower_Drill	Q 0.1	Включити подачу свердла вниз
Raise_Drill	Q 0.2	Включити подачу свердла нагору
Clamp_Workpiece	Q 0.3	Затиснути заготовку

З опису завдання випливає, що процес свердління заготовки здійснюється за допомогою трьох функціональних одиниць – пристрою затискання деталі (Vise), привода обертання свердла (Motor) і пристрою подачі свердлильної головки (Feed). Для кожної функціональної одиниці потрібний один граф станів. Для координації роботи цих функціональних одиниць потрібен ще один граф станів – свердління (Drilling).

Порядок створення графа станів можна розглянути на прикладі привода подачі «Feed».

На рисунку 1.4 показаний функціональний блок гідроциліндра подачі з діаграмами сигналів керування й зворотного зв'язка. Він містить два електромагнітних клапанів (Up і Down) і два кінцевих вимикачів (Top і Bottom).

Як видно з діаграми, процес керування гідроциліндром складається із чотирьох станів – 1, 2, 3, 4.

На діаграмі не показаний стан 0, який повинен бути в кожному графові. Він призначений для перевірки поточного положення функціонального блоку і його переключення (при необхідності) у стан готовності, коли свердлильна головка перебуває у верхньому положенні.



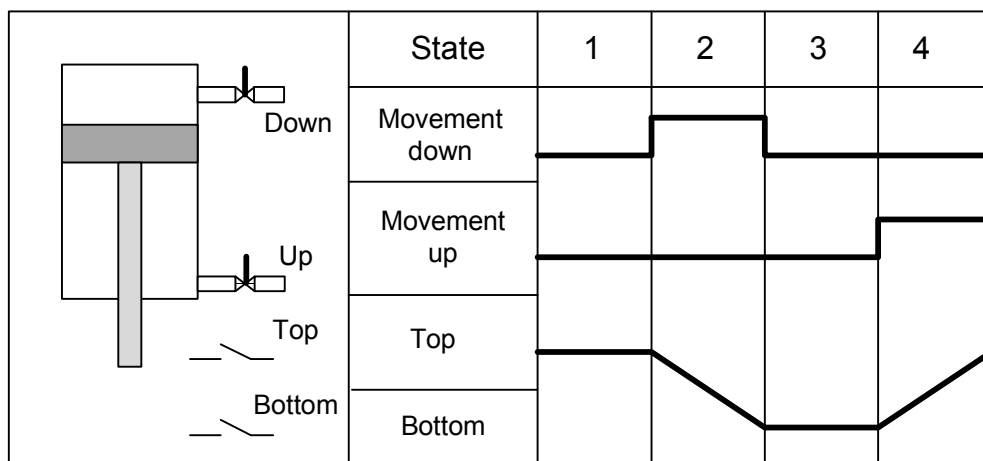


Рисунок 1.4 – Функціональний блок гідроциліндра подачі свердла

Вивід свердла у верхнє положення повинен проводитися по закінченню кожного циклу, тобто в стані 4. Якщо з якоїсь причини в момент включення верстата свердло перебувало в іншому положенні, то перехід зі стану 0 повинен бути зроблений саме в стан 4, щоб забезпечити переміщення свердла нагору.

З обліком цього складена послідовність зміни станів графа «Feed» (Подача), яка наведена на рисунку 1.5.



Рисунок 1.5 – Послідовність виконання графа станів «Feed»

Передбачається, що клапани з електромагнітним керуванням повинні використовуватися тільки для фази руху й гідроциліндр залишається у верхньому кінцевім положенні при знятті сигналів керування.

Приклад програми «Drilling\_machine» (Zen03\_02\_Higraph\_Drillmac) розташований в папці Sample projects (Типові проекти).

### 1.3 Послідовність створення графа станів в Higraph

Розглянемо деталі створення графа станів у кожному кроці.

*Крок 1. Створення проекту програми в Simatic Manager.*

Для створення нової програми необхідно відкрити Simatic Manager, у вікні, що відкрилося, «New Project» увести ім'я нового проекту й натиснути ОК.

У лівій панелі центрального вікна Simatic Manager установити курсор на імені проекту й правою кнопкою миші викликати контекстне меню. У списку, що відкрився, вибрати Insert New Object ► SIMATIC 300 Station. Далі у вікні HW-config під станцію SIMATIC 300 потрібно вставити рейку RACK 300 (Rail), блок живлення й процесорний модуль CPU315-2 PN/DP. Процес компонування елементів проекту закінчується вставкою S7 Program, яка буде відображена в правій панелі.

*Крок 2. Вставка графа станів у програму.*

Для вставки графа станів потрібно подвійним клацанням лівої кнопки миші розкрити структуру S7 Program, вибрати контейнер «Sources» і правою кнопкою викликати контекстне меню, у якому вибрати Insert New Object ► State graph. При цьому в правій панелі з'явиться об'єкт «State graph(1)», якому тут же слід привласнити ім'я, наприклад, «Motor». Після подвійного клацання по піктограмі «Motor» тільки що створений початковий граф станів буде відображений в основнім вікні редактора Higraph (рис. 1.6).

Під вікном редактора показане відкрите вікно інструкцій (Instructions), у якому можна ввести команди для програмування станів і транзакцій. За допомогою кнопок унизу вікна можна також відкрити вікна оголошення змінних (Variables) і повідомлень (Application messages і Document messages).

*Крок 3. Визначення сигналів, необхідних для керування.*

Для відображення розділу змінних виберемо в меню View опцію Details. При цьому відкриється ліва панель Environment (навколишнє середовище), у якій відображаються елементи програми, згруповані в розділи. Для перегляду змінних слід розкрити розділ Interface. У нижній частині вікна редактора перебуває вікно редагування змінних, яке відкривається кнопкою Variables. Це вікно являє собою таблицю із чотирма колонками – ім'я, тип даних, коментар і початкове значення. При цьому початкове значення не редагується, а встановлюється системою.

Вид вікна редагування після зроблених у цьому кроці налаштувань показаний на рисунку 1.7.

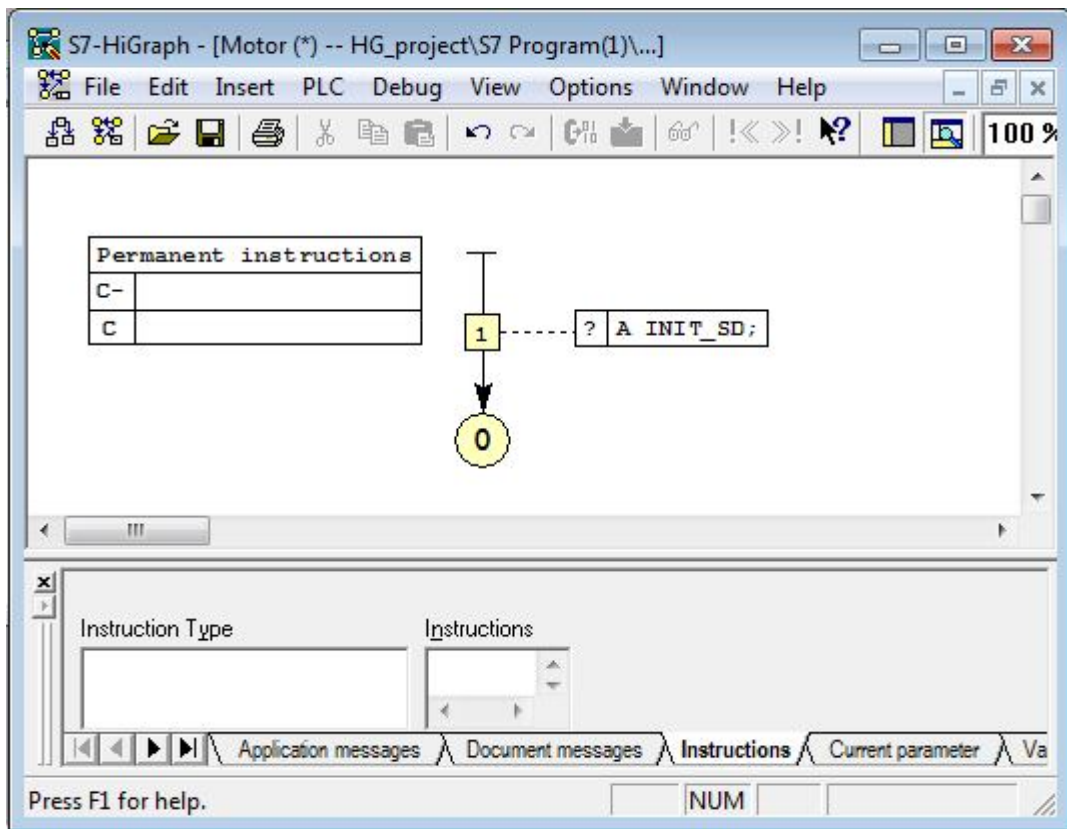


Рисунок 1.6 Вікно редактора з початковим графом станів

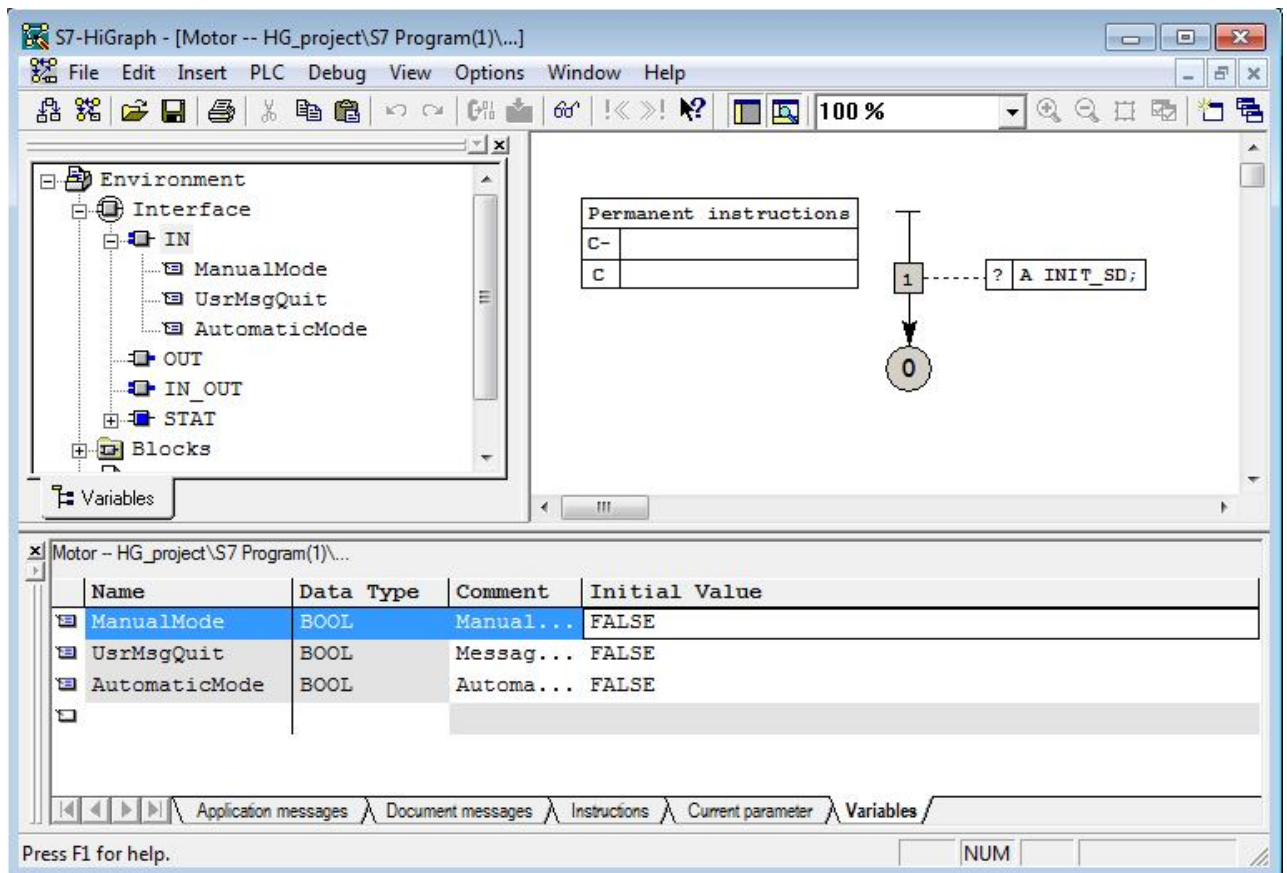


Рисунок 1.7 – Вид вікна редагування на кроці 3

Розділ змінних містить наступні значення:

- *Вхідні змінні IN.* При створенні графа редактор автоматично вводить у список вхідних змінних три змінні – Automaticmode, Manualmode і Usrmsgquit, видаляти які не можна. При цьому вхідна змінна Automaticmode при значенні «1» дозволяє обробку транзакцій тільки з атрибутом «Auto» і забороняє обробку транзакцій з атрибутом «Manual». Аналогічно, змінна Manualmode при значенні «1» дозволяє обробку транзакцій тільки з атрибутом «Manual». Змінна Usrmsgquit служить для підтвердження помилки або повідомлення.
- *Вихідні змінні OUT.* У список вихідних змінних уводяться імена вихідних параметрів графа станів.
- *Вхідні/вихідні змінні IN\_OUT.* У цей список включаються змінні для обміну повідомленнями між графами станів.
- *Змінні STAT.* Для спрощення процесу програмування редактор Nigraph автоматично встановлює в цьому списку 15 необхідних для роботи змінних, у тому числі INIT\_SD, яка забезпечує запуск програми. Значення змінних представлені в табл. 1.2.

Таблиця 1.2 – Значення змінних розділу STAT

Ім'я змінної	Значення	Тип даних	Визначається	
			Користувачем	Системою
WT_Expired	Закінчення часу очікування	BOOL		Так
WT_Valid	Час очікування активно	BOOL		Так
WT_Stop	Зупинка часу очікування	BOOL	Так	
WT_Currvalue	Збереження часу очікування	DWORD		Так
Usrmsgsend	Повідомлення стану активно	BOOL		Так
Usrmsgstat	Для внутрішнього використання	WORD		
ST_Expiredprev	Закінчення контрольного часу попереднього стану	BOOL		Так
ST_Expired	Закінчення контрольного часу	BOOL		Так
ST_Valid	Контрольний час активний	BOOL		Так
ST_Stop	Зупинка контрольного часу	BOOL	Так	
ST_Currvalue	Збереження контрольного часу	DWORD		Так
<b>INIT_SD</b>	<b>Параметр запуску</b>	<b>BOOL</b>	<b>Так</b>	
Currentstate	Номер поточного стану	WORD		Так
Previousstate	Номер попереднього стану	WORD		Так
Statechange	Зміна стану	BOOL		Так

При оголошенні змінних дозволяється наступне.

В імені змінної дозволяються текстові символи й символи підкреслення, причому символ підкреслення може стояти на початку імені, але не повинен стояти наприкінці імені.

При оголошенні типу даних редактор пред'являє на вибір BOOL, INT, WORD, CHAR, String, TIME і т.д.

Для повідомлень, які оголошуються в розділі IN\_OUT, у вікні оголошень відкривається колонка Message Type (тип повідомлення).

У розділі коментаря допускається будь-яка вистава тексту.

У вікні оголошення змінних не передбачене введення адреси. Адреса змінної, її формальний параметр, буде визначена при заповненні таблиці ідентифікаторів Symbols, яка перебуває в папці S7 Program Simatic Manager.

Відразу після того, як новий граф стану створений, змінні Currentstate, Previousstate і Statechange деактивуються.

Для активізації цих змінних необхідно:

1. Виділити змінну у вікні оголошення змінних і в контекстному меню вибрати команду Object Properties.
2. У діалогові вікні перейти на вкладку "Attributes" і призначити значення "true" на атрибут "S7\_active", як показано на рисунку 1.8.

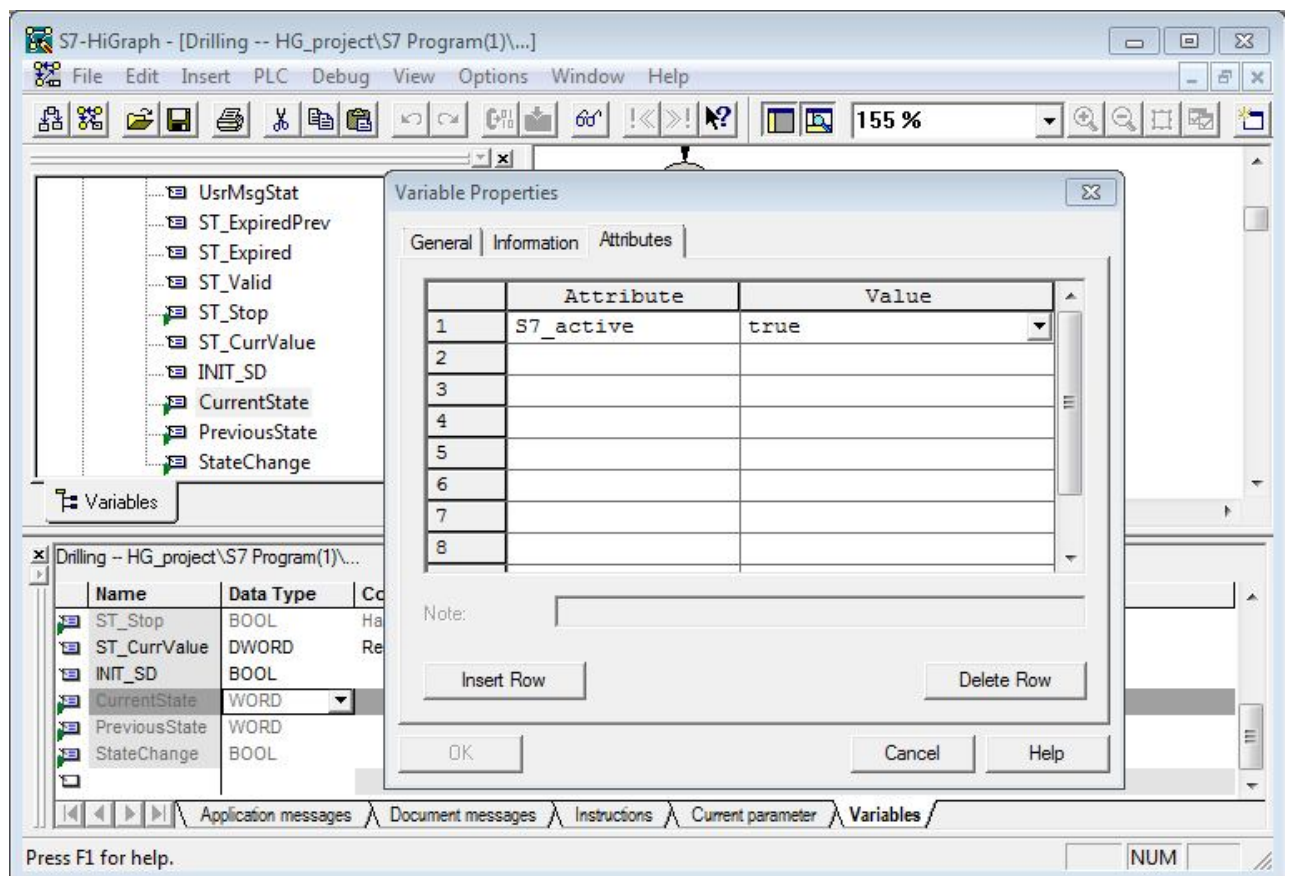


Рисунок 1.8 – Приклад активування змінної Currentstate

#### Крок 4. Програмування станів.

Програмування станів містить у собі:

- присвоєння імені стану (не обов'язково);
- введення команд;
- визначення часу очікування й контрольного часу (не обов'язково);
- вставка наступного стану.

Вставка стану проводиться командою контекстного меню Insert State. Стани нумеруються в порядку, у якому вони вводяться. Присвоєння імені стану проводиться у вікні Object Properties, виклик якого можливий після виділення стану правою кнопкою миші. У цьому ж вікні можна змінити номер стану.

Для введення команд потрібно відкрити вікно Instructions, двічі клацнувши по стану. При цьому у вікні Instructions (рис. 1.9) відобразиться список типів інструкцій. Типи інструкцій, які вводяться в стан графа, представлені в таблиці 1.3.

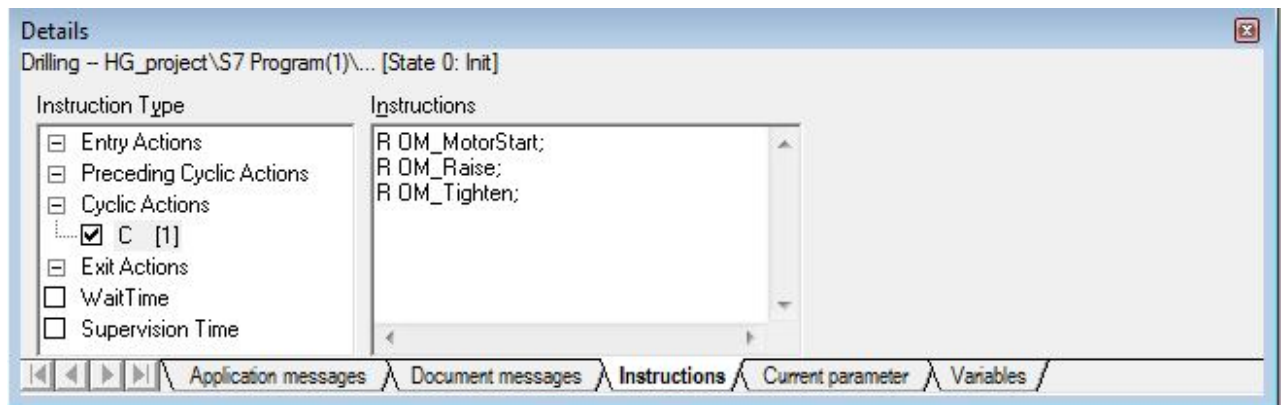


Рисунок 1.9 – Вид вікна введення команд Instructions при програмуванні стану 0

Таблиця 1.3 – Типи інструкцій, які вводяться в стані

Тип команди (інструкції)	Ідентифікатор	Опис команди
Дія входу	E	Дія, яка виконується одного разу тільки при вході в стан
Попередня циклічна дія	C-	Дія, яка містить певні умови й виконується перед перевіркою транзакції
Циклічна дія	C	Дія, яка виконується після перевірки транзакції
Дія виходу	X	Дія, яка виконується одного разу тільки при виході зі стану

Уведення інструкції проводиться в правім полі, яке стає активним після виділення типу інструкції й вибору в контекстному меню єдиної команди Insert.

При введенні STL-команди можна використовувати символічні й формальні параметри. Уведення повинно закінчуватися крапкою з комою. Результат програмування стану 0 показаний на рис. 1.10.

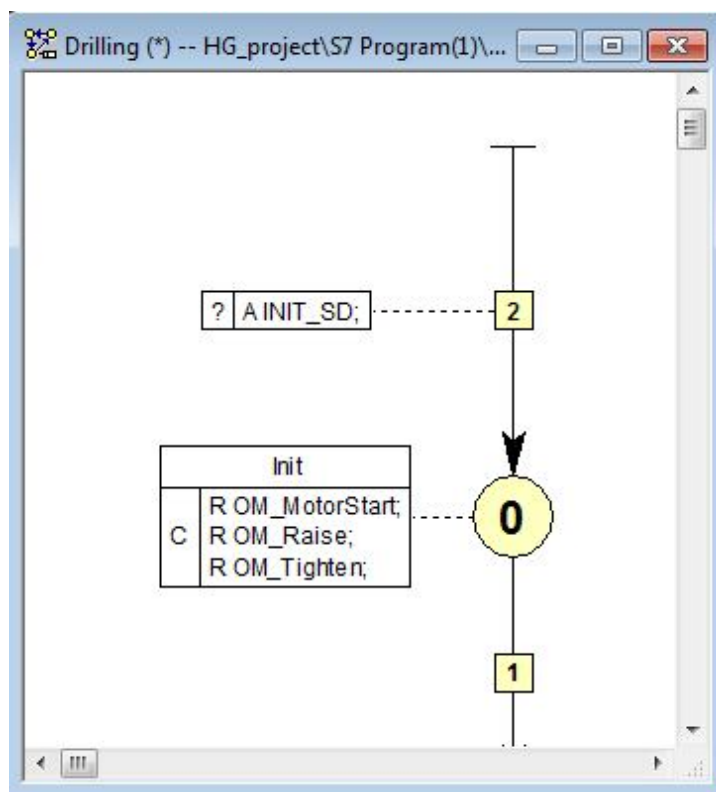


Рисунок 1.10 – Вид вікна після програмування стану 0

При програмуванні стану можна визначити, чи повинен контролер залишатися в стані якийсь час до того, як буде перевірена наступна транзакція. Установка часу очікування проводиться при виборі команди "Wait Time". При цьому в правім вікні за замовчуванням встановлюється час T#500 ms, який може бути змінений.

Якщо необхідно задати час, протягом якого процес може перебувати в стані, то слід вибрати команду "Supervision Time". Установлений за замовчуванням контрольний час 500 ms можна змінити. Якщо фактичний час перебування перевищить заданий контрольний час, то попередньо встановлена змінна "ST\_Expired" установиться в стан «1». При цьому в діагностичний буфер CPU буде видане повідомлення про помилку.

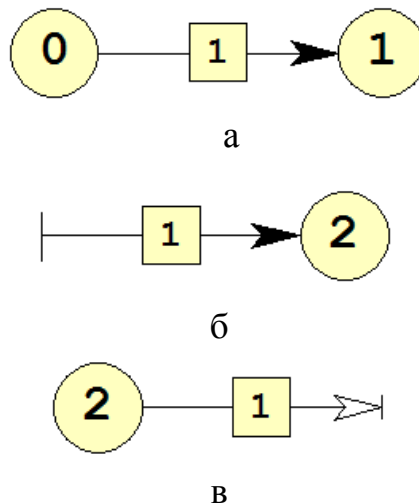
Для діагностики процесу виконання програми в стані можна призначити дві характеристики – помилка (функція F) і повідомлення (функція M). Ці призначення проводяться у вікні Object Properties. У цьому ж вікні можна призначити коментар для стану.



*Крок 5. Програмування транзакцій.*

Транзакція містить розв'язну умову для переходу від одного стану до іншого. Стану може бути призначено одна або кілька відповідних транзакцій. Якщо виконуються умови декількох транзакцій, то перемикання відбудеться по транзакції з найвищим пріоритетом (рівень 1).

У мові Ніgraph використовуються три типи транзакцій (рис. 1.11) – стандартна, довільна й транзакція повернення.



*Рисунок 1.11 – Типи транзакцій*

Стандартна транзакція (рис. 1.11,а) здійснює перехід зі стартового стану в наступний стан.

Довільна транзакція (рис. 1.11,б) іде від будь-яких станів до цільового стану. Вона має більш високий пріоритет, чим інші типи транзакцій і обробляється безупинно незалежно від поточного стану графа стану. Такі транзакції використовуються для постійного контролю важливих умов з високим пріоритетом. Якщо запрограмована в довільній транзакції контролююча функція виконується, то здійснюється перехід на галузі процесу із цільовим станом.

Транзакція повернення (рис. 1.11,в) іде від поточного стану до попереднього активного стану.

Вставка транзакції проводиться командою контекстного меню Insert Transition. Тип транзакції залежить від позиції кінця транзакції.

При програмуванні транзакцій виконуються наступні кроки:

- визначення пріоритету транзакції (не обов'язково);
- уведення умов;
- уведення дій транзакцій (не обов'язково);
- призначення імені транзакції (не обов'язково);
- уведення коментарів (не обов'язково);
- установка часу очікування (не обов'язково);



Якщо одному стану призначено кілька транзакцій, то редактор автоматично призначає цим транзакціям різні пріоритети. Бажаний рівень пріоритету можна встановити у вікні Object Properties.

Умови (Conditions) у транзакціях записуються з ідентифікатором ? (знак питання). Умови програмуються у форматі команд мови STL.

Дії в транзакціях (Transition actions) програмуються з ідентифікатором ! (знак вигуку) теж у форматі команд мови STL. Ці команди виконуються однократно, коли транзакція перемикає стан.

Для введення команд необхідно двічі клацнути по транзакції, щоб відкрити вікно редагування. Далі в лівій частині вікна редагування вибрати зі списку Conditions або Transition actions і ввести STL-команду в правій області вікна. Уведені команди відображаються у вікні графа станів, як таблиця, «прикріплена» до транзакції.

Слід урахувати, що обробка команд починається з результатом логічної операції  $RLO = 1$ .

#### *Крок 6. Програмування постійних інструкцій.*

Постійні команди виконуються один раз у цикл, незалежно від поточного стану. У постійних командах можна програмувати наступні основні процеси:

- Обчислення змінних процесу.
- Реєстрація й обробка подій, на які процес повинен завжди реагувати, незалежно від поточного стану, наприклад, контроль захисту й блокувань.

Для редагування доступні наступні типи постійних команд:

- Циклічні дії (Preceding Cyclic Actions), які завжди виконуються на початку циклу (ідентифікатор C-).
- Циклічні дії (Cyclic Actions), які завжди виконуються наприкінці циклу (ідентифікатор C).

Щоб програмувати ці дії, потрібно клацнути по таблиці команд із заголовком "Permanent Instructions". При цьому відкриється вікно редагування, у лівій області якого необхідно вибрати тип інструкції, а в правій увести команду STL. Додавання команд проводиться так: у лівій області потрібно виділити тип інструкції, потім натиснути праву кнопку й вибрати єдину команду Insert.

Після введення всіх команд вони будуть відображатися у вікні графа стану, як таблиця.

#### *Крок 7. Програмування операційних режимів.*

Якщо в транзакції запрограмувати режим Manual (ідентифікатор M), то транзакція буде перемикатися тільки в ручному режимі, а якщо запрограмувати режим Auto (ідентифікатор A), то транзакція буде перемикатися тільки в автоматичному режимі. Необхідна особливість режиму встановлюється у вікні Object Properties. Після установки режиму транзакція зафарбовується в

рожевий (Auto) або блакитний (Manual) колір і забезпечується відповідним ідентифікатором (А або М).

#### *Крок 8. Створення координуючого графа.*

З функціональної діаграми, представленої на рисунку 1.3, видно, що процес свердління складається з 8 станів.

Стан 1 характеризує початкову позицію (ініціалізацію) – лещата розціплені, свердло у верхньому положенні й мотор виключений. У цій позиції можливі установка й зняття заготовки.

Перехід у стан 2 здійснюється пусковою кнопкою (Start\_Button). Коли процес затискання закінчиться й спрацює тензодатчик зусилля затискача (Tension\_Reached), транзакція перемкне процес у стан 3. У цьому стані запускається мотор обертання свердла й по досягненню заданої швидкості (сигнал Drill\_Motor\_running) відбувається перехід у стан 4 – включення привода подачі свердла.

Подача проводиться до моменту спрацювання кінцевого вимикача в крайньому нижньому положенні свердлильної головки (стан 5). У цьому положенні свердло повинне обертатися без подачі якийсь час для зменшення пружних деформацій від осьової подачі свердла й потім автомат повинен перейти в стан 6, де реалізується команда руху свердлильної головки нагору. По закінченню процесу, коли спрацює кінцевий вимикач крайнього верхнього положення головки, автомат перейде в стан 7. Тут відбувається вимикання мотора й після його зупинки (сигнал з датчика швидкості Drill\_Motor\_stopped) здійснюється перемикання в стан 8, у якому проводиться розтиск лещат.

На рисунку 1.12 показаний координуючий граф станів Drilling, який відповідає функціональній діаграмі процесу свердління.

### **1.4 Вимоги до звіту по роботі**

Звіт по роботі повинен містити наступні матеріали:

1. Вихідні дані для розробки програми (завдання).
2. Роздруківки всіх графів станів (первинних файлів) і таблиці символів.
3. Описи створених графів станів.

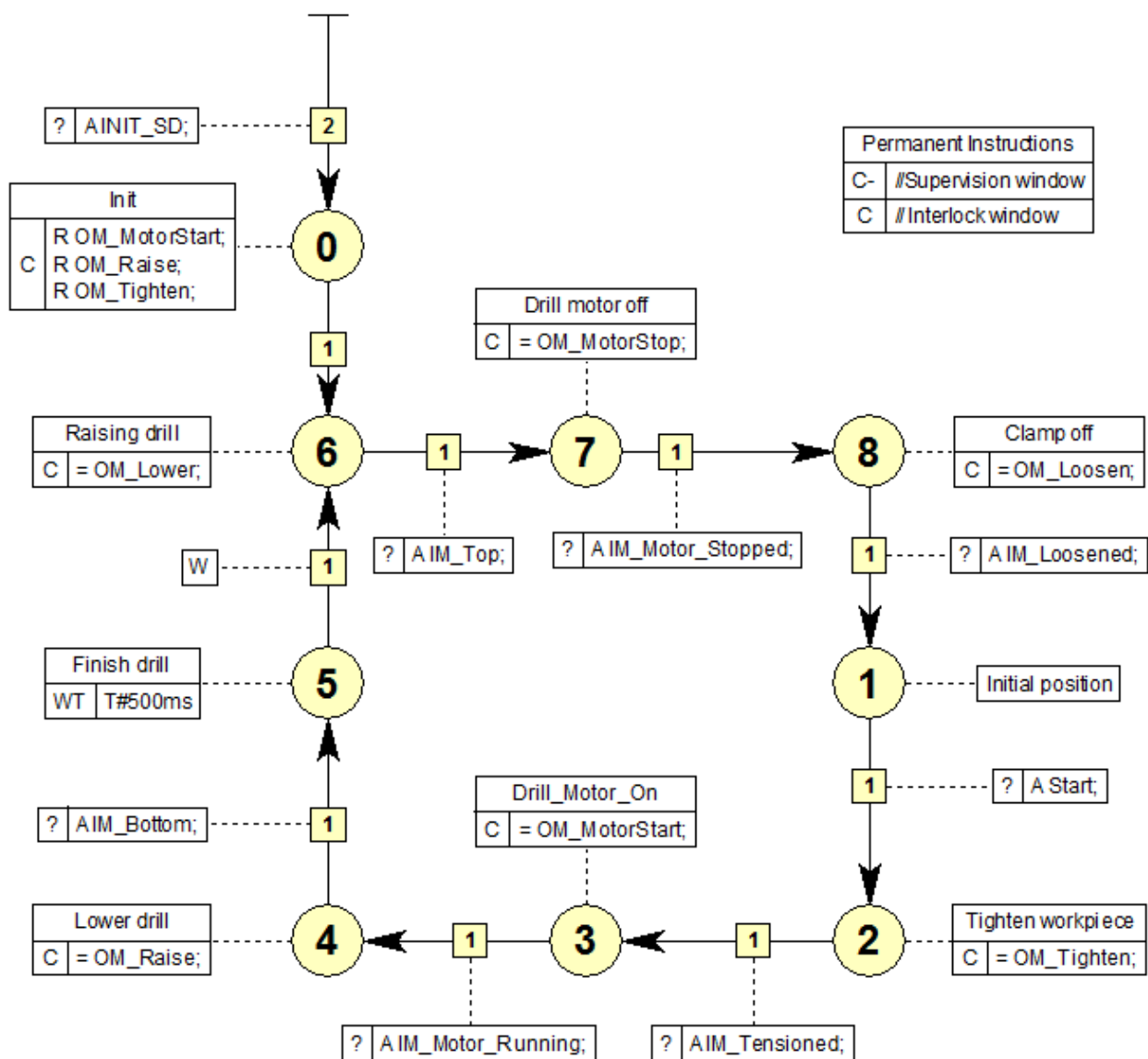


Рисунок 1.12 – Вид координуючого графа станів для свердильного верстата

## 2 РОЗРОБКА Й НАЛАГОДЖЕННЯ ПРОГРАМИ Higraph

*Ціль роботи: освоєння приймань і придбання навичок створення програм керування встаткуванням у редакторі S7-Higraph – інструментальному додатку програмної системи STEP 7.*

### 2.1 Створення групового графа

Груповий граф визначає задану послідовність запитів до графів станів, які виконуються циклічно. Запит до графа стану відомий як запит до первиннику (первинному файлу). Первинники обробляються в програмувальному контролері як змінні групового графа області STAT.

Для створення групового графа потрібно виконати наступне.

В Simatic Manager відкрити папку Source Files із графами станів. Потім клацнути на порожньому місці правою кнопкою й у контекстному меню вибрати команду Insert New Object ► Graph group. Вставленому груповому графові слід привласнити ім'я, наприклад, Drilling\_machine. Присвоєння імені здійснюється у вікні Object Properties, яке відкривається за допомогою контекстного меню.

Створити груповий граф можна у вікні S7-Higraph. Для цього потрібно відкрити будь-який граф станів і вибрати в меню File ► New Graph Group.

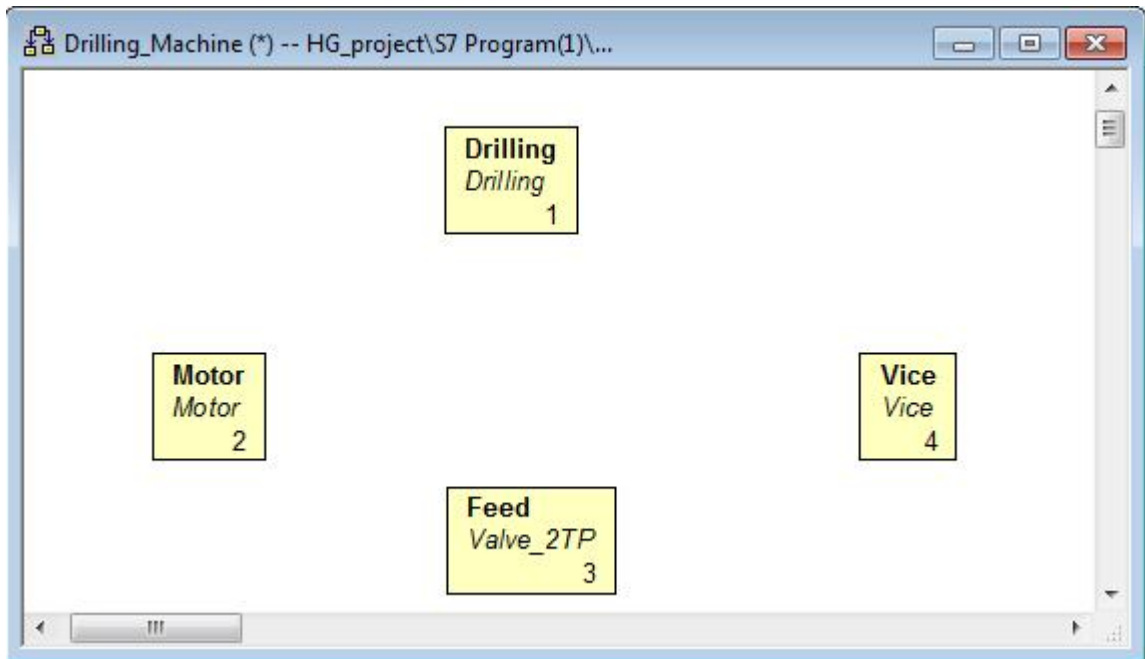
*Вставка первинників (графів станів).*

Створений груповий граф відкривається з порожньою робочою областю. Для вставки в цю область графів станів потрібно клацнути правою кнопкою на порожньому полі й у контекстному меню вибрати **Insert Instance**. При цьому відкриється вікно вибору файлу “Open”, у якому відображаються всі створені до цього графи станів. Першим вибирається координуючий граф. Закривши вікно “Open” кнопкою ОК, потрібно вставити цей граф на робоче поле групового графа. Вставлений граф станів відображається у вікні групового графа прямокутником з іменем і номером. Імена вставлених графів станів відображаються, як змінні, в області оголошення STAT.

Операції вставки потрібно повторити для всіх створених графів станів. На [рисунку 2.1](#) показане вікно групового графа із вставленими графами станів, створеними для розглянутого тут прикладу.

*Установка послідовності виконання процесу.*

Послідовність запуску графів станів задається у вікні Run Sequence, яке відкривається однойменною командою контекстного меню. Визначити цю послідовність можна по координуючому графу станів. Для розглянутого прикладу прийнята така послідовність виконання: Drilling (1), Motor (2), Feed (3), Vice (4).



*Рисунок 2.1 – Вид вікна групового графа «Drilling\_Machine» із вставленими графами станів*

*Призначення фактичних параметрів.*

Графи станів для окремих функціональних одиниць являють собою первинники, які можна вставляти в будь-які проекти програм. Зазвичай розроблювач графа дає змінним ті імена, які відбивають сутність елементарного процесу керування, наприклад, для змінної «Включити мотор» призначає ім'я “Motor\_On”. Однак, при створенні програми керування конкретним устаткуванням, у якому є кілька моторів, буде потрібно інша система іменування змінних. Тому в груповому графові передбачене зв'язування первинних імен змінних вставлених графів станів зі змінними, які використовуються в створюваній програмі. Процедура зв'язування називається *призначенням поточних параметрів*. Вона зводиться до наступного.

У вікні групового графа знаходяться вставлені графи станів (первинники). У меню View вибираємо команду Details, яка відкриває в нижній області екрана вікно редагування. Це вікно містить вкладку "Current parameters". Виділяємо один із графів стану й відкриваємо цю вкладку з відображенням списку всіх змінних для обраного графа станів. Далі відкриваємо таблицю ідентифікаторів командою Options ► Symbol Table і робимо оголошення всіх вхідних і вихідних змінних із вказівкою їх фактичних адрес і типів даних. Приклад заповнення таблиці представлений на рисунку 2.2.

Для того, щоб призначити нове ім'я змінної (це ім'я визначене в таблиці символів), потрібно на вкладці "Current parameters" вибрати ім'я в стовпці Name, а в стовпці Current parameters правою кнопкою викликати команду контекстного меню Insert Symbol/Message.

Status	Symbol	Address	Data type	Comment
1	Clamp_Workpiece	Q 0.3	BOOL	Clamp/hold workpiece with set tension
2	CYCL_EXC	OB 1	OB 1	
3	DB_GG_Drilling	DB 1	DB 1	DB for drilling graph group
4	Drill_at_Bottom	I 0.2	BOOL	Limit switch for "drill at lowest position"
5	Drill_at_Top	I 0.3	BOOL	Limit switch for "drill at highest position"
6	Drill_Motor_On	Q 0.0	BOOL	Switch on drill motor
7	Drill_Motor_Runn...	I 0.0	BOOL	Feedback signal for "drill running at set speed"
8	Drill_Motor_Stop...	I 0.1	BOOL	Feedback signal for "drill stationary"
9	GG_Drilling	FC 1	FC 1	FC for drilling graph group
1	Lower_Drill	Q 0.1	BOOL	Lower drill via feed to lowest limit
1	Raise_Drill	Q 0.2	BOOL	Raise drill via feed to highest limit
1	Start_Button	I 0.7	BOOL	Start button for drilling machine
1	Tension_Reached	I 0.4	BOOL	Feedback signal for "workpiece set tension reached"
1				

Рисунок 2.2 – Приклад заповнення таблиці Symbols

При цьому відкриється список змінних таблиці Symbols, у якому потрібно вибрати відповідну змінну й натиснути Enter. Ця процедура відображена на рисунку 2.3.

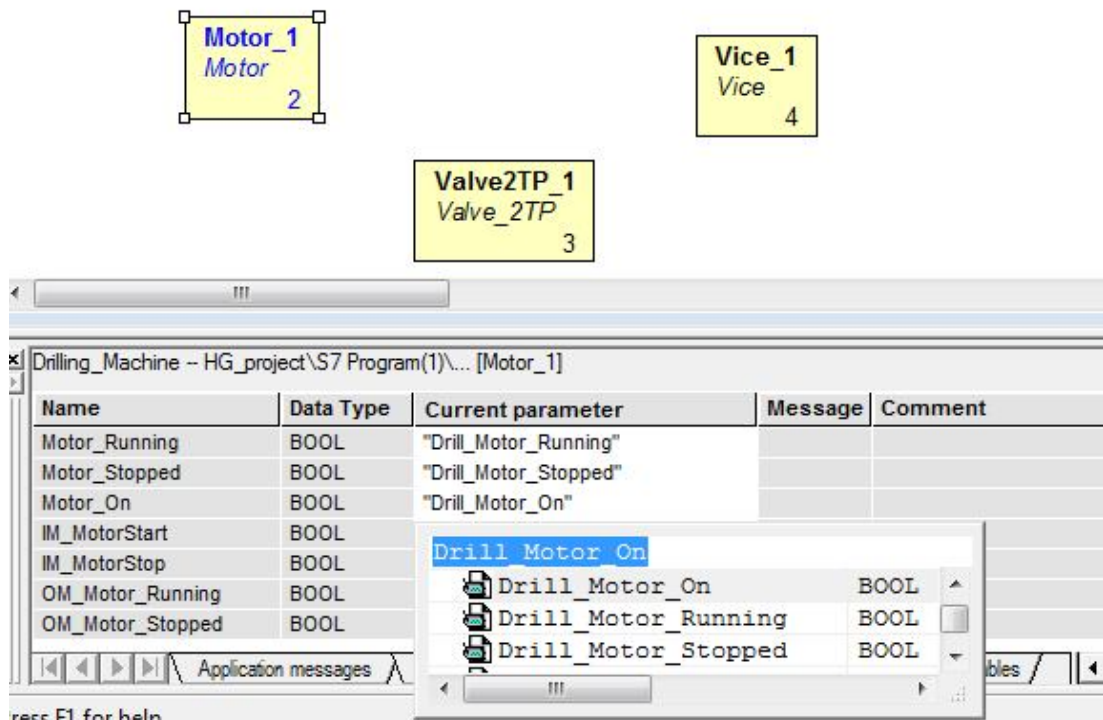


Рисунок 2.3 – Відображення процедури заміни імені змінної

Призначення поточних параметрів необхідно виконати для всіх змінних у всіх графах станів, причому в стовпець "Current parameters" потрібно внести навіть ті імена, які не мають відмінності від стовпця "Name". Нові поточні параметри не призначаються тільки для змінних типу "in".

### *Програмування повідомлень.*

Повідомлення – це бінарна змінна, яка встановлюється графом станів передавачем і обробляються в транзакціях графа станів приймача. У транзакціях приймача програмується також дія, яка скидає біт отриманого повідомлення. Повідомлення служать засобами зв'язку між графами станів і використовуються для координації взаємодії графів стану в груповому графові.

Залежно від області дії використовуються два типи повідомлень:

*Internal message* – внутрішнє повідомлення для зв'язку між графом станів і груповим графом. Зв'язок здійснюється через бітову адресу блоку даних DB групового графа. Цей вид повідомлень використовується при створенні програми керування з одним груповим графом.

*External message* – зовнішнє повідомлення для зв'язку між графами станів, що перебувають у різних групових графах, або між Nigraph-функціями FC і іншими програмами. Зв'язок здійснюється через загальнодоступну бітову адресу, установлену програмістом.

У якості повідомлень використовуються булеві змінні, оголошені як Message Type. При оголошенні змінних разом з іменем слід вказати ознаки – вихідне повідомлення позначити OM (output message), вхідне повідомлення позначити IM (input message).

Для пояснення механізму обміну повідомленнями розглянемо приклад. Нехай зі стану 3 (рис. 1.5) графа-передавача “Feed” у координуючий граф Drilling передається *вихідне* повідомлення OM\_Bottom (досягнуте дно). Поточний параметр цього повідомлення записується у вікні Current parameter групового графа з адресою й іншим типом, тобто так: Drilling.IM\_Bottom.

Координуючий граф Drilling ухвалює це *вхідне* повідомлення й використовує його в транзакції перемикачів зі стану 4 у стан 5 (рис. 1.12), де передбачена логічна операція «I» між цим повідомленням і умовою переходу, що перебуває в акумуляторі (команда: A IM\_Bottom). Якщо RLO цієї операції буде рівний «1», відбудеться перехід у стан 5, з якого після закінчення часу очікування W буде виконане перемикачів в стан 6. Тут процес піде в іншому напрямку – уже координуючий граф Drilling повинен передати *вихідне* повідомлення OM\_Lower, поточним параметром якого буде: Feed.IM\_Lower. Граф Feed ухвалює це *вхідне* повідомлення й використовує його в транзакції перемикачів в стан 4 (команда A IM\_Lower).

Таким чином, *вихідні* повідомлення відправляються зі станів, а *вхідні* використовуються в транзакціях.

Вхідні (in) і вихідні (out) повідомлення призначаються у вікні оголошення змінних Variables групового графа в стовпці Message Type. На рисунку 2.4 показані вхідні й вихідні повідомлення для графа станів Motor.



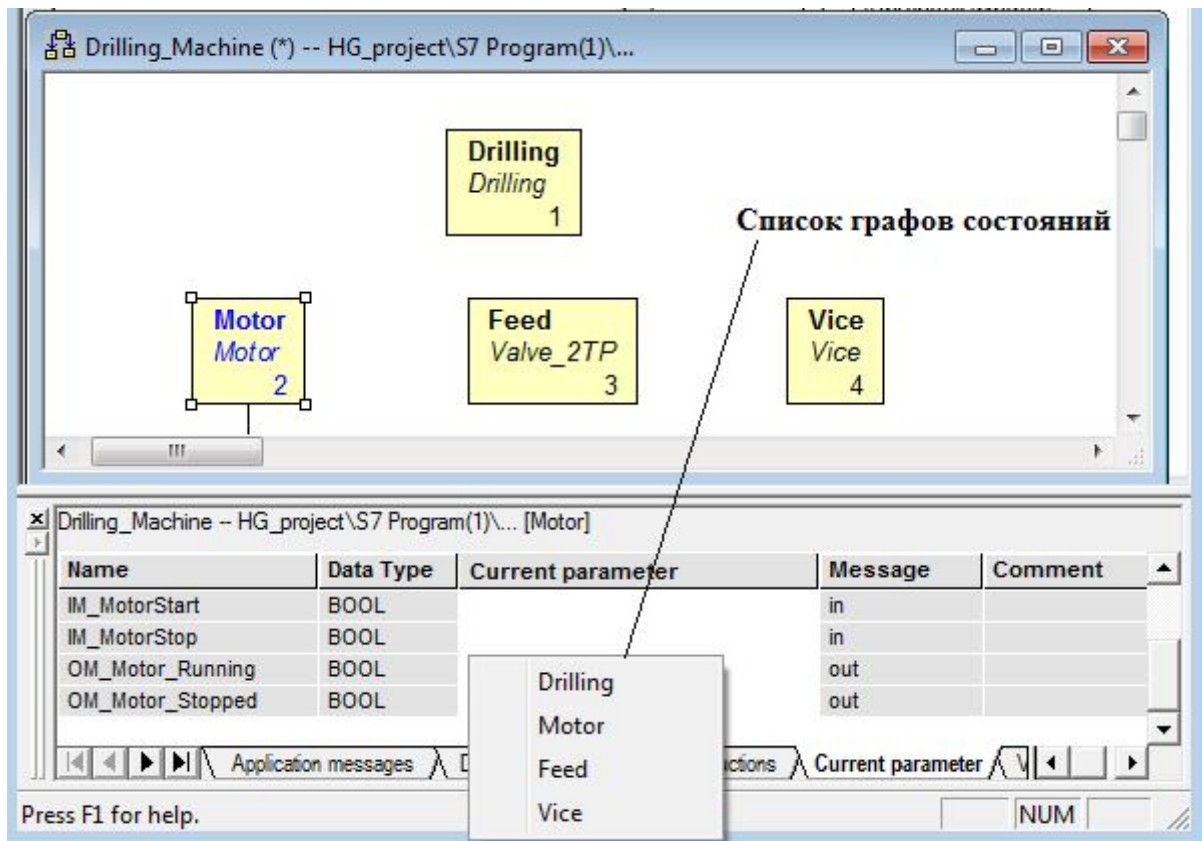


Рисунок 2.4 – Вид вікна оголошення змінних із входними й вихідними повідомленнями

Слід урахувати, що для змінних типу OUT програмуються дії присвоєння, наприклад, = OM\_Motorstart (рис.1.12, стан 3) або установки, наприклад, S OM\_Motorstart, а для входних змінних типу IN програмуються умови, наприклад, A IM\_Bottom (перехід у стан 5).

Призначення поточних параметрів для вихідних повідомлень проводиться у вікні Current parameter групового графа. Процес призначення здійснюється в такий спосіб.

Спочатку необхідно виділити граф станів. Далі у вікні оголошення змінних прокрутити список змінних і знайти вихідні повідомлення з типом «out». На перетинанні імені повідомлення й стовпця «Current parameter» клацанням правої кнопки викликати контекстне меню, у якому вибрати команду Insert Symbol/Message. По цій команді відкривається невелике вікно зі списком графів, вставлених у груповий граф. На наведеному вище рисунку 2.4 показаний список графів для передачі вихідного повідомлення OM\_Motor\_Running.

Враховуючи те, що повідомлення OM\_Motor\_Running повинне відправлятися в координуючий граф станів Drilling, клацаємо по імені Drilling і воно вставляється редактором у гніздо таблиці. Далі додаємо до імені крапку й редактор автоматично виводить список змінних для призначення цьому вихідному повідомленню поточного параметра (рис. 2.5).



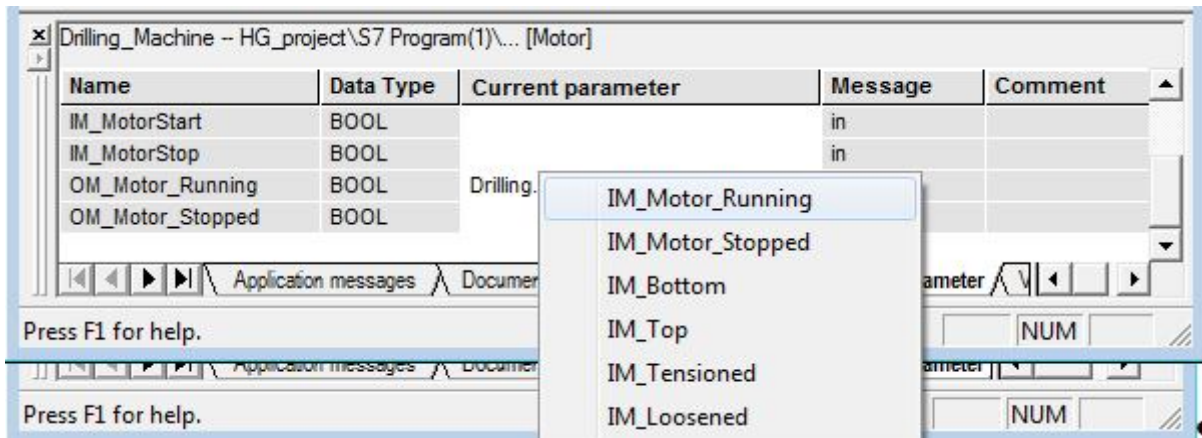


Рисунок 2.5 – Вид вікна для призначення поточного параметра

Далі вибираємо зі списку IM\_Motor\_Running і одержуємо результат – Drilling.IM\_Motor\_Running. Після призначення всіх повідомлень груповий граф приймає вид, показаний на рисунку 2.6.

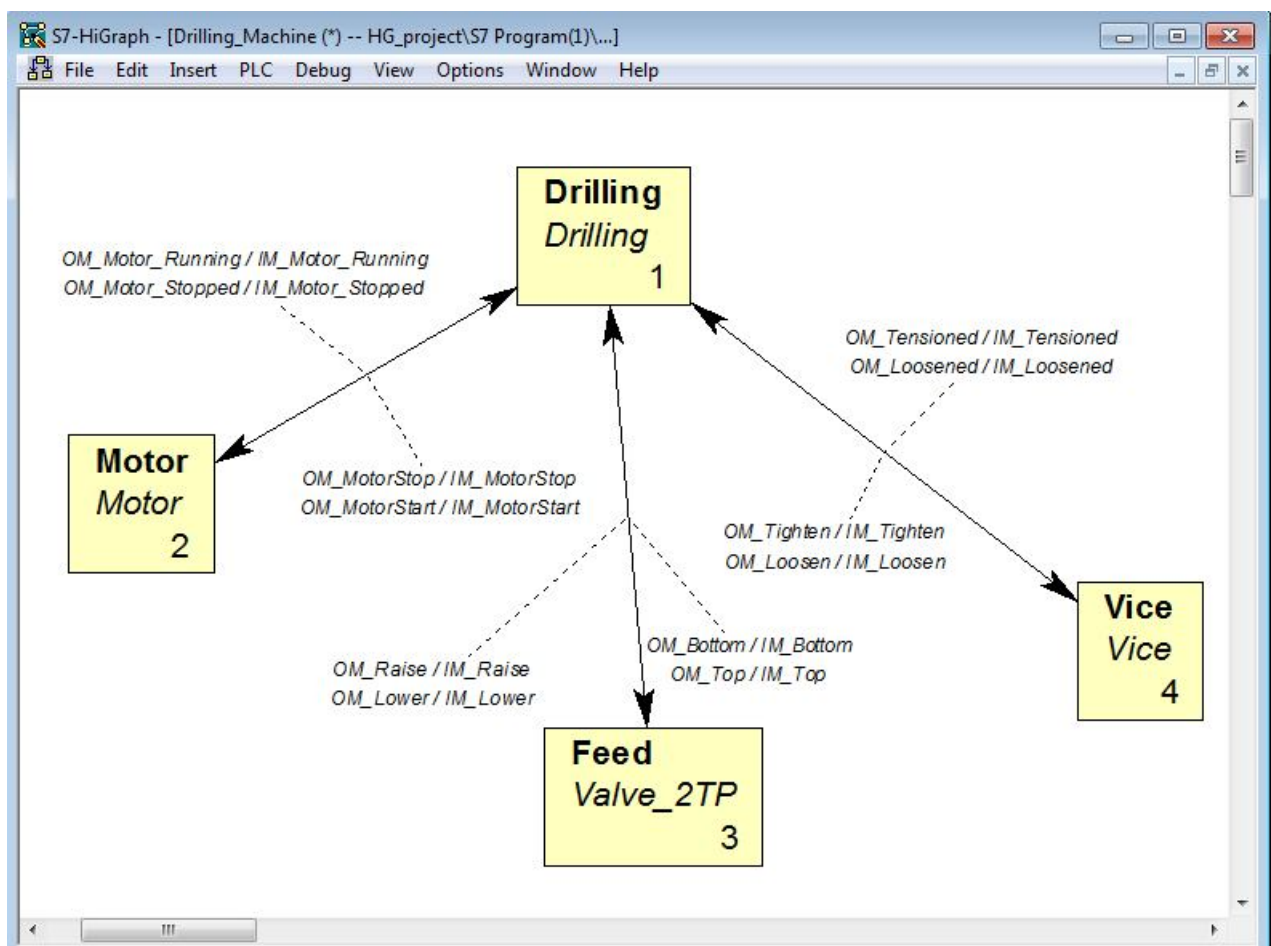


Рисунок 2.6 – Вид групового графа після призначення всіх повідомлень

## 2.2 Компіляція первинників і створення блоків програми

При збереженні графів станів у контейнері "Source Files" програми S7 ніякої перевірки синтаксису не проводиться. Тому процес редагування може проводитися протягом будь-якого числа сеансів роботи. По закінченню роботи зберегти об'єкти можна командою File > Save.

Процес компілювання застосовується тільки для групових графів, а індивідуальні графи станів не компілюються. При компіляції груповий граф спочатку зберігається, потім Nigraph перевіряє синтаксис програми, створює функцію (FC) і блок даних (DB), а потім запам'ятовує їх у контейнері "Blocks" програми S7.

Якщо при компіляції виявлені синтаксичні помилки, то вони будуть відображені у вікні повідомлення. При цьому блок не буде створений. Якщо після перевірки синтаксису з'являться тільки попередження, то логічний блок буде створений.

Для компіляції групового графа необхідно виконати наступне.

У меню Options вибрати команду «ім'я\_групового\_графа Settings». У вікні, що відкрилося, перейти на закладку Compile (рис. 2.7) і ввести ім'я функції (FC) і блоку даних (DB). При призначенні імені можна вказувати абсолютне ім'я, наприклад, FC 1. Тут же можна встановити інші параметри настроювання.

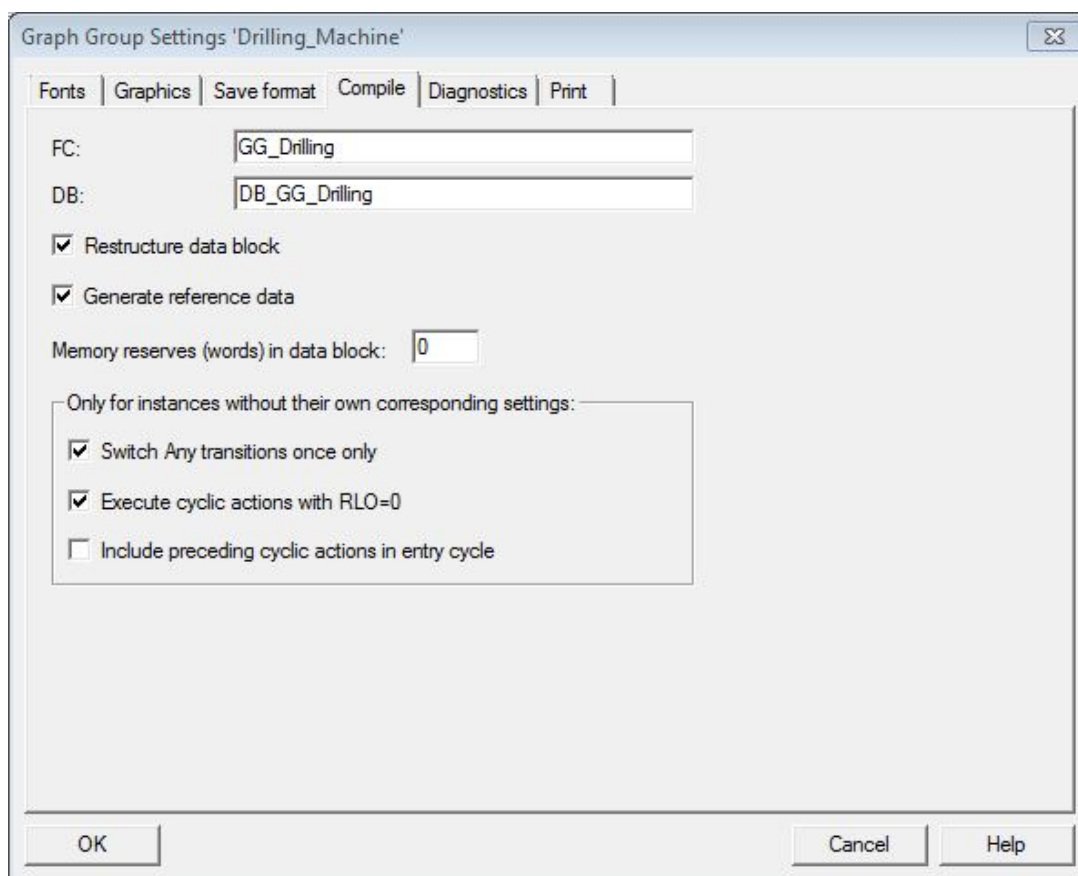


Рисунок 2.7 – Вид вікна настроювання на закладці Compile

Опції настроювань забезпечують наступні ефекти:

- Restructure data block – блок даних DB створюється в процесі трансляції.
- Generate reference data – довідкові дані генеруються автоматично.
- Memory reserves (words) in data block – установлюється резерв для додаткових графів станів і повідомлень.
- Switch Any transitions once only – опція запобігає повторному перемиканню транзакції запуску.
- Execute cyclic actions with RLO=0 – опція змушує виконати циклічні дії до виходу зі стану при RLO=0.
- Include preceding cyclic actions in entry cycle – опція змушує виконати попередні перед входом у стан дії (C-).

Після настроювань на вкладці Compile слід перемкнутися на вкладку Diagnostics, установити прапорець на опцію «Format converter diagnostics» і закрити вікно.

Для переходу до процесу компіляції потрібно вибрати команду меню File ► Compile. Процес компіляції відображається у вікні повідомлень (нижня область екрана). При виводі помилок потрібну позицію можна знайти, двічі клацнувши на повідомленні про помилки. Після виправлення помилок слід перекомпілювати груповий граф. На рисунку 2.8 показаний вид вікна групового графа з результатами компіляції.

Для циклічної обробки програми Higraph у програмувальному контролері вона повинна викликатися з організаційного блоку OB1.

Блок OB1 можна програмувати в LAD/STL/FBD редакторі базового пакета STEP 7. Функція (FC), створена в Higraph, має вхідний параметр "INIT\_SD". Цей параметр повинен установлюватися в "1" при включенні контролера, інакше графи станів у груповому графові ініціюватися не будуть. Сигнал установки "INIT\_SD" може бути сформований з використанням стартової інформації OB1 (змінна #OB1\_SCAN\_1) і збережений у тимчасовій змінній OB1, наприклад, в #startup.

Коли блок OB1 вставляється в програму, необхідні змінні вже оголошені в розділі TEMP і залишається дописати цей розділ стартовою змінною #startup (рис. 2.9).

Програма організаційного блоку OB1 містить у собі генерацію біта запуску, виклик функції FC1 і ініціалізацію змінної INIT\_SD:

```
L OB1_SCAN_1      // Генерація біта запуску.  
L 1  
==I  
= #startup        //Установка першого циклу OB1.  
CALL "GG_Drilling" //Виклик функції FC1.  
INIT_SD:=#startup //Ініціалізація сигналу установки.
```

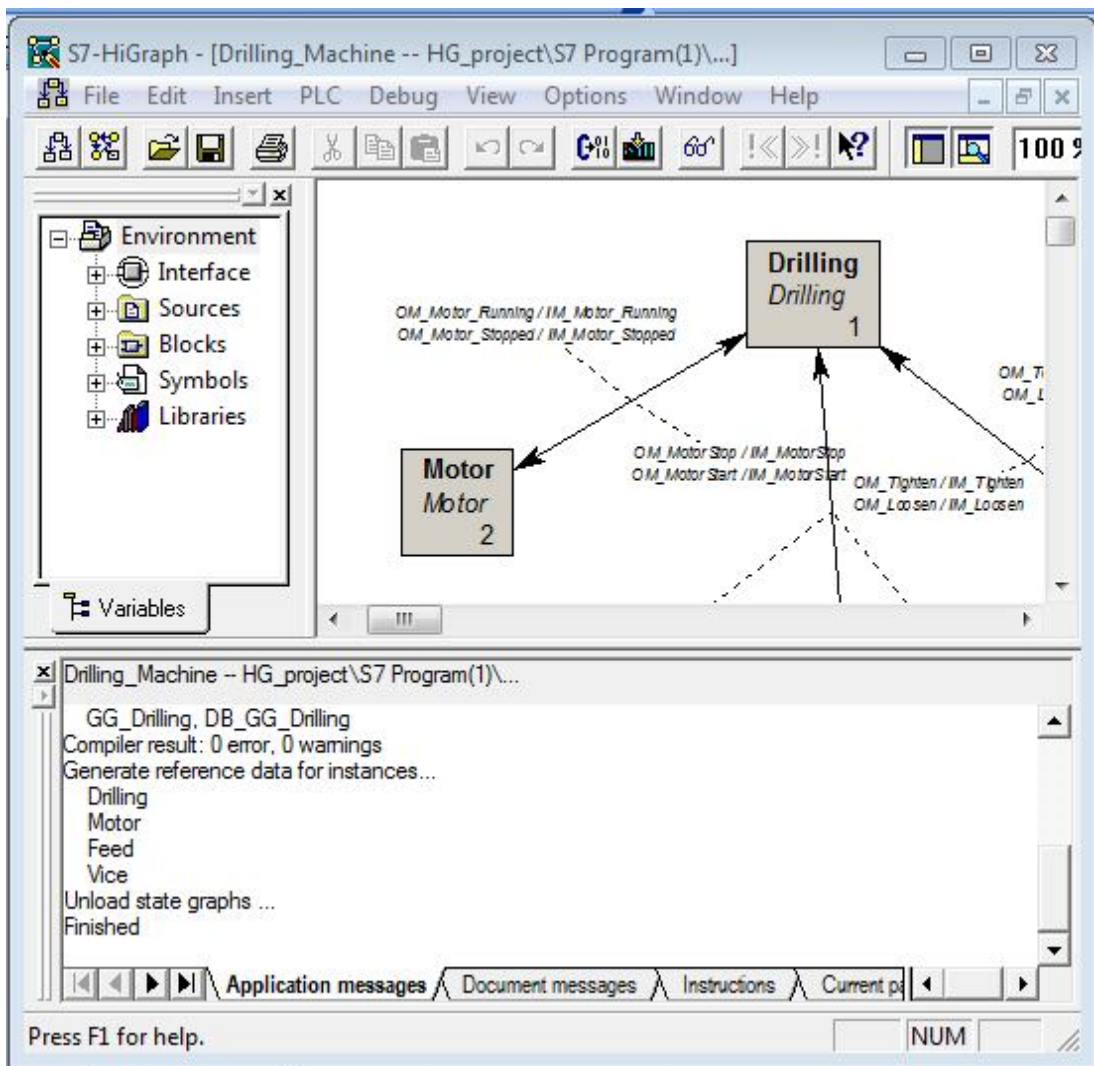


Рисунок 2.8 – Вид вікна групового графа з результатами компіляції

		Contents Of: 'Environment\Interface\TEI		
	Name	Data Type	Address	
Interface	OB1_EV_CLASS	Byte	0.0	
TEMP	OB1_SCAN_1	Byte	1.0	
OB1_EV_CLASS	OB1_PRIORITY	Byte	2.0	
OB1_SCAN_1	OB1_OB_NUMBR	Byte	3.0	
OB1_PRIORITY	OB1_RESERVED_1	Byte	4.0	
OB1_OB_NUMBR	OB1_RESERVED_2	Byte	5.0	
OB1_RESERVED_1	OB1_PREV_CYCLE	Int	6.0	
OB1_RESERVED_2	OB1_MIN_CYCLE	Int	8.0	
OB1_PREV_CYCLE	OB1_MAX_CYCLE	Int	10.0	
OB1_MIN_CYCLE	OB1_DATE_TIME	Date...	12.0	
OB1_MAX_CYCLE	Startup	Bool	20.0	
OB1_DATE_TIME				
Startup				

Рисунок 2.9 – Розділ оголошення змінних блоку OB1

## 2.3 Завантаження програми в контролер і її налагодження

Для завантаження програми користувача в контролер, повинні бути виконані наступні вимоги:

- Програма, яка буде завантажуватися, повинна бути відкомпільована без помилок.
- Повинен бути запрограмований виклик Higraph FC із циклічно виконуваного блоку.
- Пристрій програмування й програмувальний контролер повинні бути з'єднані.

Завантаження програми здійснюється в наступній послідовності:

1. Установити CPU у режим STOP.
2. Відкрити Higraph програму в SIMATIC Manager.
3. Вибрати необхідні блоки в контейнері блоків:
  - Higraph FC;
  - Higraph DB;
  - блок виклику (OB або FB);
  - Higrapherremitterfb (FB20), якщо потрібна діагностика;
  - Higraphmsgemitterfc (FC101), якщо потрібна діагностика.
4. Вибрати команду меню PLC > Download.

Для завантаження в програмувальний контролер тільки функції (FC) з пов'язаним з нею блоком даних (DB) необхідно при відкритому груповому графові вибрати команду меню PLC > Download і в діалогові вікні "Download" вибрати завантаження блоку даних DB разом з функцією FC.

Існує можливість контролю й зміни програми, у той час як вона виконується в CPU. Це дозволяє знайти помилки, які не були відображені формальною логікою перевірки при створенні програми або при перевірці синтаксису протягом трансляції.

Редактор Higraph дозволяє виявити наступні помилки:

- Програмні помилки, наприклад, неправильно встановлений контрольний час.
- Логічні помилки в структурі програми, наприклад, запрограмовані стани й транзакції не відповідають необхідній послідовності технологічних операцій.

Слід урахувати, що функція налагодження сповільнює проходження програми й може привести до збоїв або перевищення часу циклу.

Доступні наступні функції налагодження й контролю:

- Контроль стану програми.
- Контроль і зміна значень змінних.
- Оцінка довідкових даних.

Перш, ніж використовувати контролюючі функції, повинні бути виконані наступні вимоги:

- Пристрій програмування повинен бути інтерактивно пов'язаний з CPU.
- Програма, повинна бути відкомпільована без помилок.
- Програма (включаючи FC, DB, і OB) повинна бути завантажена в CPU.
- CPU повинен бути в режимі RUN (читання) або в режимі RUN-P (читання й запис).
- Ніграф FC повинна викликатися із блоку OB1.

Просування програми через індивідуальні стани й транзакції, а також поточна інформація щодо оброблюваних команд показується на екрані. Вікна Ніграф надають наступні можливості контролю:

- У вікні групового графа видні стани й усі первинники групового графа, причому поточний стан відображений у кожному графові.
- У вікні графа станів активний стан позначений кольором, а транзакція, яка привела до цього стану і попередній активний стан затінений.

Щоб запуснути контроль стану програми, необхідно:

1. При відкритому груповому графові вибрати команду меню Debug ► Monitor для відображення стану групового графа.
2. Вибрати один або декілька первинників, а потім команду меню Edit ► Open Object. Кожний обраний первинник буде відкритий інтерактивно. Відображається детальна інформація стану.
3. Таблиця з інформацією стану відображається спочатку для переміщення з найвищим пріоритетом, що веде від активного стану. Якщо потрібно, можна вибрати іншу активну таблицю команд, щоб відобразити її інформацію.
4. Щоб вибрати для контролю інші первинники, потрібно повернутися до короткого огляду стану групового графа, вибрати інші первинники й знову застосувати команду меню Edit ► Open Object.
5. Щоб вийти з відображення стану програми, слід дезактивувати команду меню Debug ► Monitor.

Для зміни змінних потрібно вибрати команду меню Debug ► Select Variable. У діалогові вікні, що відкрилося, потрібно вибрати необхідні первинники і їх змінні, а після редагування натиснути кнопку "OK".

При налагодженні програми можна використовувати різні довідкові дані. Для їхнього перегляду слід вибрати команду Options ► Reference.

## **2.4 Порядок виконання роботи й вимоги до звіту**

При виконанні роботи слід створити груповий граф, для якого призначити поточні параметри й повідомлення. Після цього зробити компіляцію створеного групового графа й програмування організаційного блоку OB1.

Результати цієї роботи представляються у звіті, який повинен містити роздруківки групового графа, скомпільованої таблиці символів, програми блоку OB1, а також структури блоків S7-програми.

Завантаження програми в контролер для перевірки її працездатності й налагодження може бути виконане тільки в тому випадку, коли первинники ретельно пророблені (оцінка за першу роботу більш 90 балів по 100-бальній шкалі).

## 3 КОНФІГУРУВАННЯ СИСТЕМИ ЧПУ SINUMERIK

*Ціль роботи: освоїти методику конфігурування систем ЧПУ й підготовки специфікації апаратних засобів.*

### 3.1 Інтерфейс конфігуратора NCSD

Конфігуратор NCSD є інструментом швидкого проектування систем ЧПУ. Він являє собою платформу для формування специфікації компонентів, необхідної для замовлення апаратних засобів.

Інтерфейс конфігуратора наведено на рисунку 3.1.

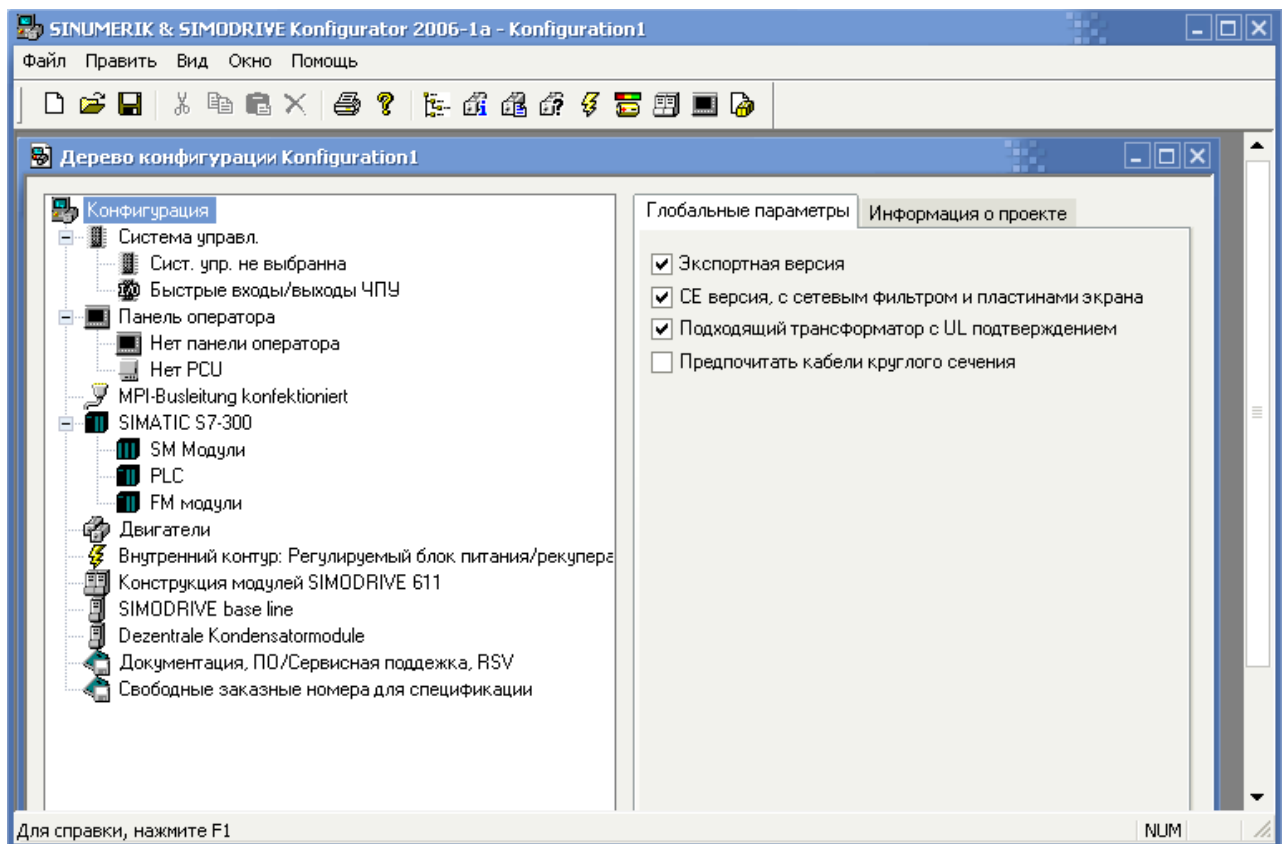


Рисунок 3.1 – Вид вікна конфігуратора NCSD

У лівій частині вікна конфігуратора представлено дерево проекту, а в правій – зміст виділеного пункту.

Кожний пункт проекту може містити кілька вкладок. Для конфігурування системи ЧПУ необхідно послідовно пройти по всіх пунктах проекту, роблячи необхідні операції на вибір компонентів і забезпеченню необхідних параметрів системи.

Перший пункт проекту називається “Конфігурація”. На вкладці “Глобальні параметри” потрібно вибрати експортну версію системи з



мережним фільтром. На вкладці “Інформація про проект” можна ввести дані розроблювача й найменування проекту.

Після заповнення цих двох форм переходимо до наступного пункту – “Система керування”. Тут необхідно вибрати виконання системи ЧПУ. Для вибору системи пропонується ряд варіантів (рис. 3.2).

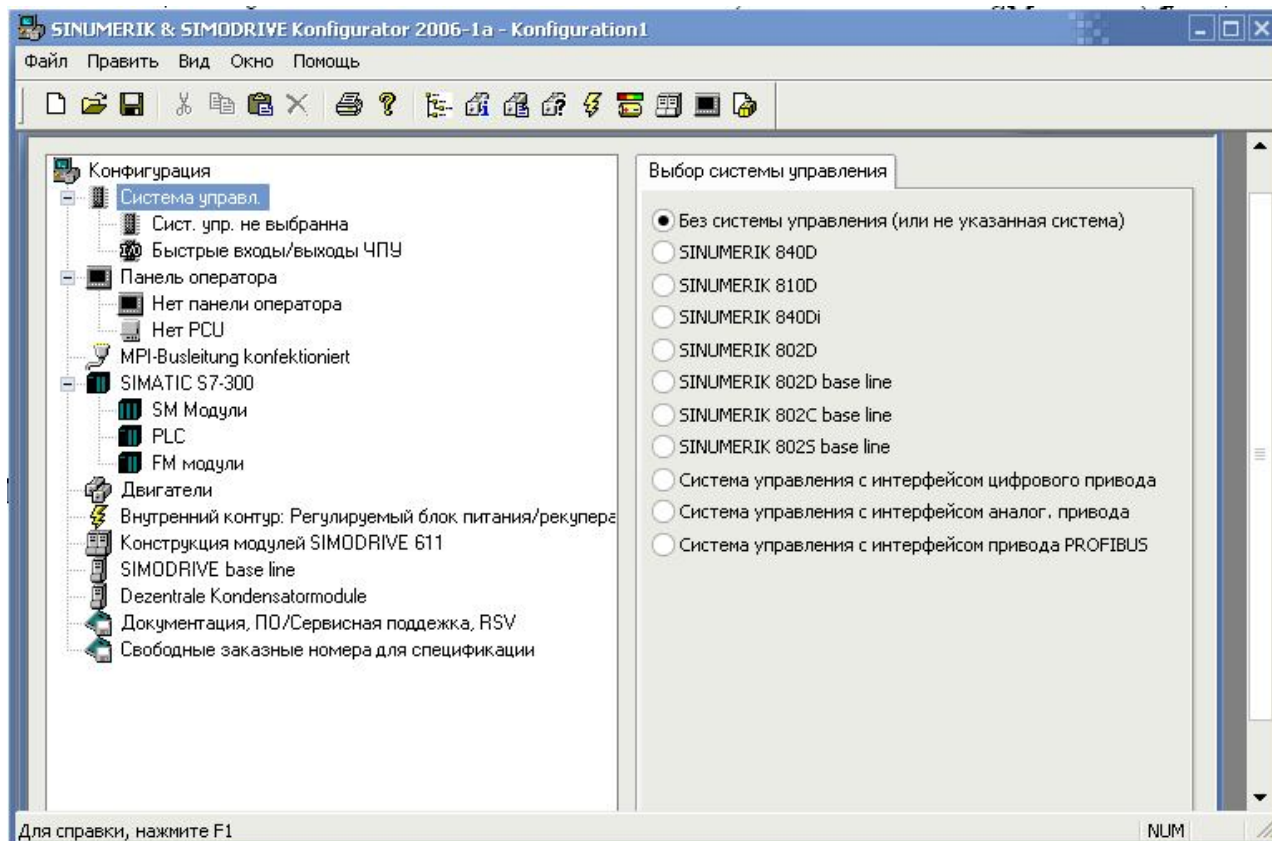


Рисунок 3.2 – Варіанти вибору виконання системи

При виборі системи ЧПУ слід керуватися призначенням кожної системи.

Так, система SINUMERIK 802D являє собою один блок PCU (Panel Control Unit), призначений для розв'язку простих завдань – керування чотирма осями й одним шпинделем. Для керування автоматикою передбачений модуль PP 72/48, а для керування приводами – перетворювач SIMODRIVE 611 universal E із аналоговим каналом.

Система SINUMERIK 810D поєднує в одному модулі CCU (Compact Control Unit) усі завдання ЧПУ, програмувального контролера й комунікації. CCU перебуває в одному корпусі з інтегрованими силовими модулями, що забезпечують роботу із двома приводами подачі й одним приводом шпинделя. З метою розширення кількості осей (максимально до 6) використовуються спеціальні плати розширення й перетворювачі SIMODRIVE 611 із цифровим інтерфейсом.

Система SINUMERIK 840D – це системна платформа, на якій можна реалізувати найбільш складні завдання керування. У комбінації із програмувальним контролером SIMATIC S7-300 і перетворювачами SIMODRIVE 611 digital вона дозволяє створювати комплекси програмного керування із числом керованих осей від 2 до 31.

Слід урахувати, що при утрудненні у виборі виконання системи на вкладці “ Система керування” можна зробити вибір одного із трьох останніх варіантів, у яких представлені можливі схеми керування двигунами.

Нехай, наприклад, кращим є варіант “ Система керування з інтерфейсом цифрового привода”. Вибравши цей варіант, переходимо до наступного пункту проекту – вибір панелі оператора, а після цього вертаємося до пункту “Система керування” і продовжуємо вибір уже в обмеженому діапазоні – для вибору залишилися тільки дві системи: SINUMERIK 810D і SINUMERIK 840D (рис. 3.3).

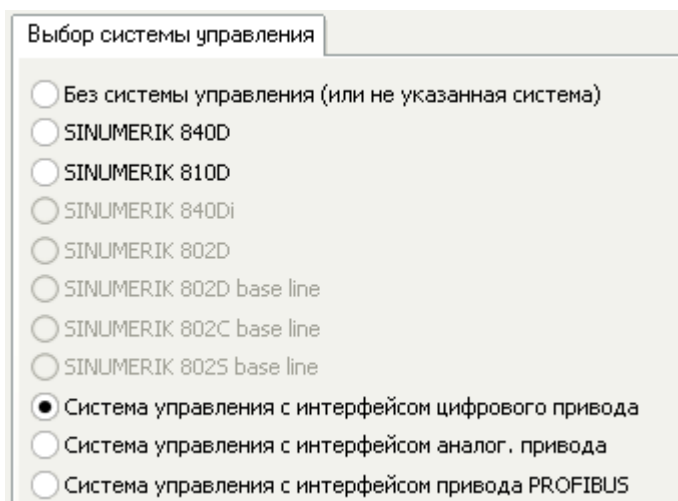


Рисунок 3.3 – Вид вкладки “Вибір системи керування” після завдання інтерфейсу цифрового привода

Якщо обрана, наприклад, система керування SINUMERIK 840D, то, клацнувши в дереві конфігурування на наступному рядку, переходимо до вибору модуля керування NCU (Numeric Control Unit) – пристрою числового керування.

У базовій конфігурації система комплектується NCU 561.4, який може виконувати тільки прості завдання керування по двом координатам – лінійній осі та осі шпинделя. При цьому лінійна вісь виконує тільки позиціонування, а інтерполяція реалізується у вигляді зв'язку двох осей (для нарізання різьблення). Для керування більш складними процесами необхідно встановити модулі розширення NCU, перераховані на вкладці “NCU” (рис. 3.4).

Вибір NCU проводиться з урахуванням кількості необхідних груп режимів роботи (ГРП), кількості каналів програмування й кількості

підтримуваних осей. Слід урахувати, що нормальна конфігурація передбачає одну групу режимів роботи. У пропонованій на рис. 5.4 таблиці в дужках наведена максимальна кількість груп, які можуть бути створені в цій системі.

NCU	Опции ЧПУ 1	Опции ЧПУ 2	
	ГРР	Каналы	Осей
<input checked="" type="radio"/> Нет NCU			
<input type="radio"/> NCU 561.4	1	1 (2)	1 (2)
<input type="radio"/> NCU 561.5	1	5 (6)	5 (6)
<input type="radio"/> NCU 571.4	1	5 (6)	5 (6)
<input type="radio"/> NCU 571.5	1 (10)	1 (10)	5 (31)
<input type="radio"/> NCU 572.4	1 (10)	1 (10)	5 (31)
<input type="radio"/> NCU 572.5	1 (10)	1 (10)	5 (31)
<input type="radio"/> NCU 573.4	1 (10)	1 (10)	5 (31)
<input type="radio"/> NCU 573.5	1 (10)	5 (31)	5 (31)

Рисунок 3.4 – Вид вкладки “NCU” для вибору пристрою числового керування

Одна група режимів роботи поєднує канали ЧПУ осями й шпинделями в один оброблюваний блок, тобто являє собою багатоканальний пристрій ЧПУ. У середині групи кожна вісь може програмуватися в будь-якому каналі. Якщо канал один, а осей 5, то керування осями проводиться послідовно в одному робочому циклі.

Після вибору системи керування й панелі оператора переходимо до конфігурування контролера для керування автоматикою.

На цьому етапі спочатку потрібно визначити тип центрального процесора контролера. Для вибору пропонується три типи – CPU 314, CPU 315-2 DP і CPU 317-2 DP. Вони відрізняються швидкодією, обсягом робочої пам'яті, у якій виконується користувацька програма, і кількістю підтримуваних каналів вводу-виводу. Тому, приступаючи до вибору CPU, необхідно мати уявлення про приблизний обсяг користувацької програми, а також про кількість вхідних і вихідних сигналів (аналогових і дискретних).

Нехай, наприклад, обраний CPU 315-2 DP. Тоді в наступному пункті меню “SM модулі” виводиться вкладка з полями для заповнення кількості входів і виходів. На рисунку 3.5 показаний приклад заповнення полів цієї вкладки.

При переході в наступний пункт конфігурування проекту “PLC” конфігуратор виводить механічну конфігурацію контролера SIMATIC S7-300 з однієї або двома стійками, у яких установлені необхідні модулі. На цій вкладці потрібно визначити довжину сполучного кабелю між головної стійкою й стійкою розширення. Вибираємо довжину кабелю з урахуванням відстані між шафами, наприклад, 2,5 м і переходимо до пункту “FM модулі”.

Входы/Выходы    Список модулей вх./вых.

Количество входов/выходов

Входов		Выходов		
Цифровы	24 В DC	40	24 В DC, 0.5 А	45
	120 В AC	12	120 В AC, 0.5 А	15
			24 В DC, 2 А	10
			120/130 В AC, 2 А	5
			Реле	6
Аналоговых		4		4

Разрешить 32-контактные модули

Комбинированный модуль 16вх+16вых цифровых:

Комбинированный модуль 8вх+8вых цифровых:

Комбинированный модуль 4вх+2вых аналоговых:

Выбор модулей EFP  ▼

Модули EFP в конфигурации

Рисунок 3.5 – Приклад заповнення вкладки "Входи-Виходи" для пункту проекту "SM модулі"

Функціональні модулі FM використовуються для керування кроковими двигунами (FM 353) і серводвигунами (FM 354) у завданнях позиціонування. При цьому кроковий двигун управляється через силовий модуль FM STEPDRIVE, а серводвигун – через інтерфейс  $\pm 10\text{В}$  аналогового перетворювача SIMODRIVE 611. У випадку вибору модуля FM 353 необхідно буде вказати тип модуля STEPDRIVE, що залежить від крутного моменту крокового двигуна.

Заключний етап конфігурування пов'язаний з вибором двигунів, перетворювачів і датчиків.

При переході в пункт меню "Двигуни" виводиться вкладка "Вибір двигуна" зі списком двигунів (рис. 3.6). На цій же вкладці необхідно вибрати в списку, що розкривається, тип перетворювача.

Процедура вибору двигуна спрощується при натисканні на кнопку "Помічник". Помічник виводить ряд додаткових вікон, у яких необхідно вказати технологічну схему, номінальну швидкість і номінальний момент, а також типи датчиків.

Після виконання нескладних процедур в окремій вікні виводиться оптимізований список рекомендованих двигунів з більш детальною інформацією (рис. 3.7). Вибравши конкретний тип, завершуємо процедуру натисканням на кнопку "Готове".

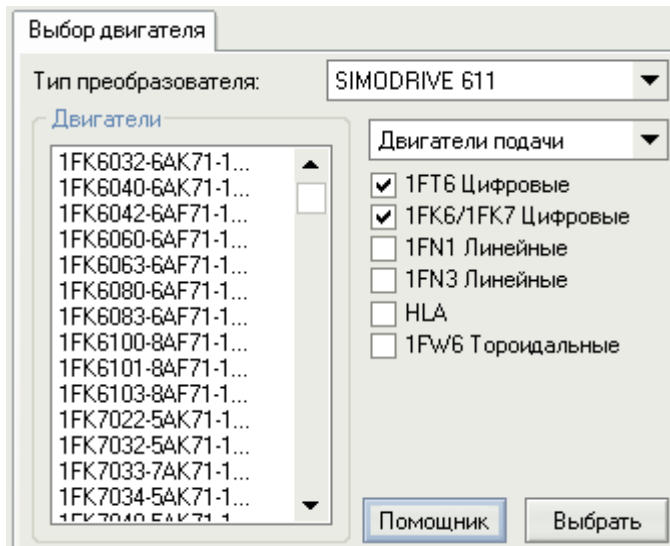


Рисунок 3.6 – Вивід списку двигунів на вкладці “Вибір двигуна”

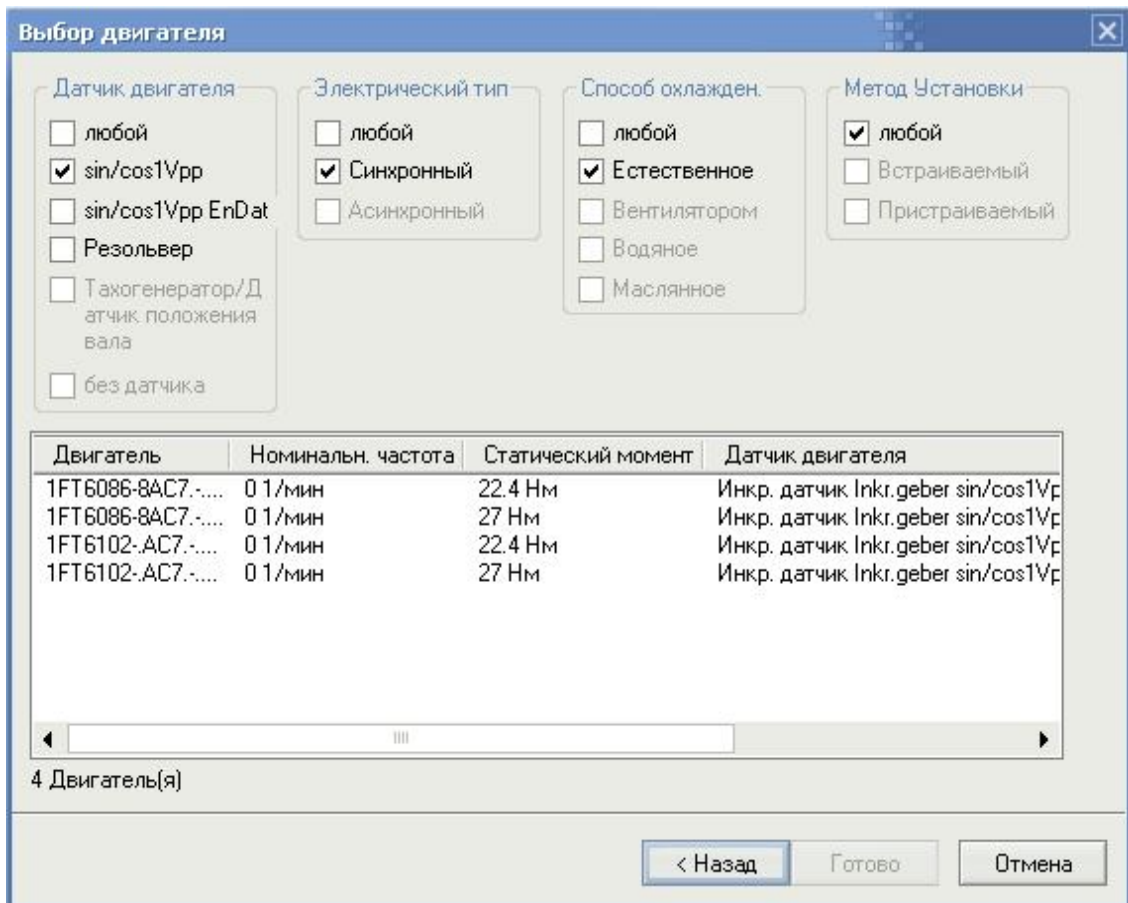


Рисунок 3.7 – Вид додаткового вікна “Вибір двигуна”

У такій же послідовності робимо вибір інших двигунів. При цьому конфігуратор робить **автоматичну комплектацію** двигунів необхідними перетворювачами й модулями живлення (рис. 3.8).

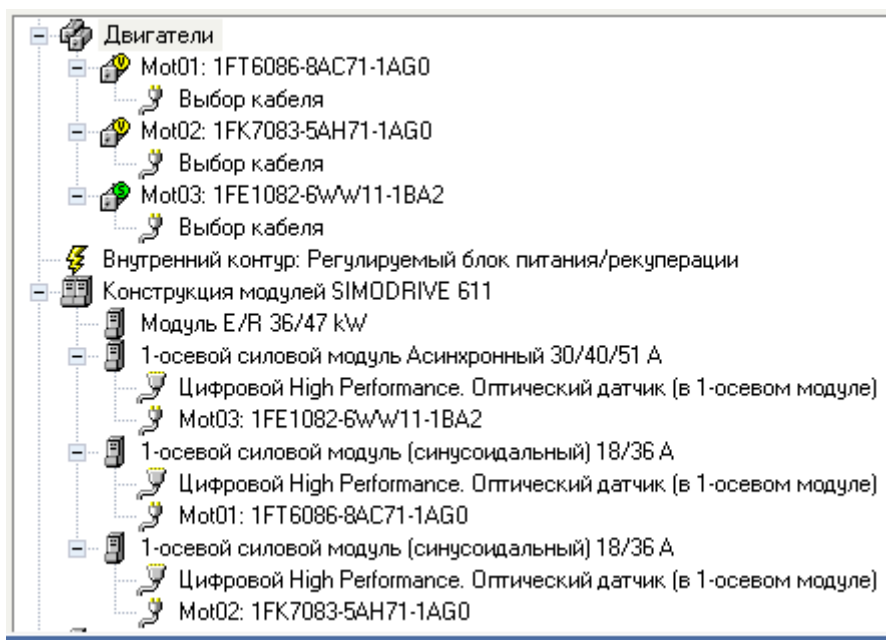


Рисунок 3.8 – Фрагмент дерева проекту, що відображає двигуни й перетворювачі SIMODRIVE 611 із цифровими каналами керування

Для документування проекту необхідно виконати наступне:

1. Зберегти файл конфігурації з розширенням “.ncsd”, вибравши команду “Файл” ⇒ “Зберегти як”.
2. Зберегти файл специфікації компонентів проекту з розширенням “.xml”, вибравши для цього команду “Файл” ⇒ “Експорт специфікації” ⇒ “Без силових і сигнальних кабелів”.
3. Роздрукувати файл специфікації, вибравши команду “Файл” ⇒ “Попередній перегляд” ⇒ “Специфікація”.

За результатами конфігурування можна одержати також інформацію про конструкцію системи, сумарному навантаженню й характеристиках двигунів, Для цього меню містить відповідні кнопки.

### 3.2 Приклад конфігурування системи ЧПУ

#### **Завдання:**

Конфігурувати систему ЧПУ фрезерного верстата із трьома лінійними осями (X, Y, Z) і шпинделем.

*Вимоги до панелі оператора:* дозвіл 800x600 пікселів, SVGA.

*Вимоги до контролера:*

- обсяг програми користувача – близько 100 Кбайт.
- кількість дискретних входів 24В – 50;



- кількість дискретних входів 120В – 20;
- кількість дискретних виходів 24В (0,5А) – 35;
- кількість дискретних виходів 120В (0,5А) – 15;
- кількість дискретних виходів на реле – 18;
- кількість аналогових входів – 4;
- кількість аналогових виходів – 4;
- кількість модулів простої периферії – 1.

*Вимоги до приводів:*

- крутний момент двигунів подачі – 25 Нм;
- потужність привода головного руху – 15 кВт;
- тип двигуна подачі – синхронний серводвигун;
- тип двигуна привода шпинделя – асинхронний.

*Вимоги до системи живлення двигунів:*

Передбачається застосування цифрових каналів керування перетворювачами. Схеми живлення повинні бути постачені модулями рекуперації, а двигуни захищені від перенапруги.

### **Конфігурування системи**

Результати конфігурування системи ЧПУ показано на рисунку 3.9.

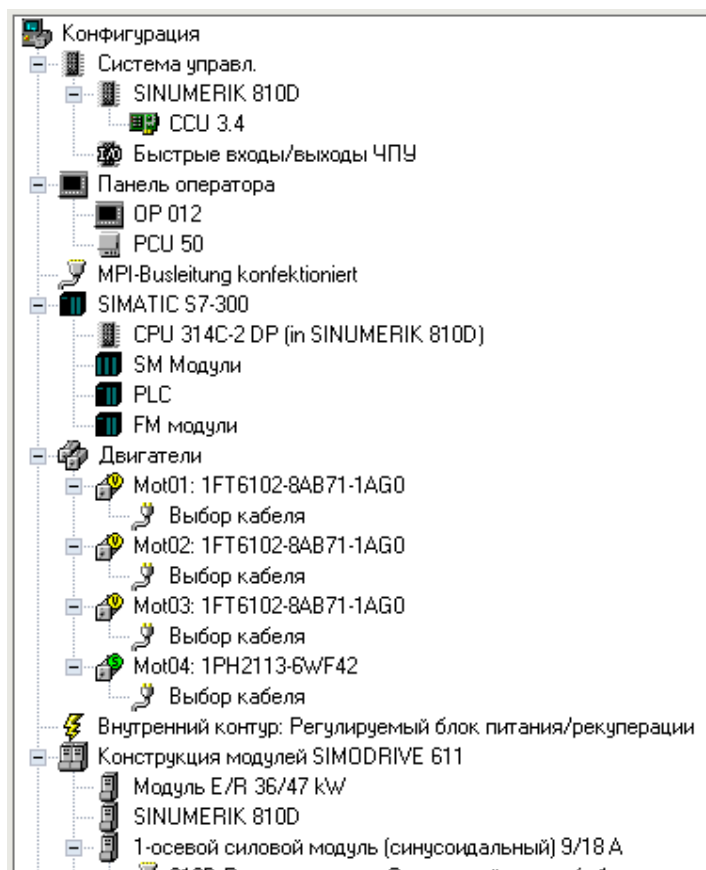


Рисунок 3.9 – Вид дерева проекту для системы ЧПУ фрезерного верстака

Увесь хід прикладу конфігурування системи можна переглянути у файлі *Konfiguration\_Primer.ncsd*.

Конструктивне зображення системи ЧПУ показано на рисунку 3.10.

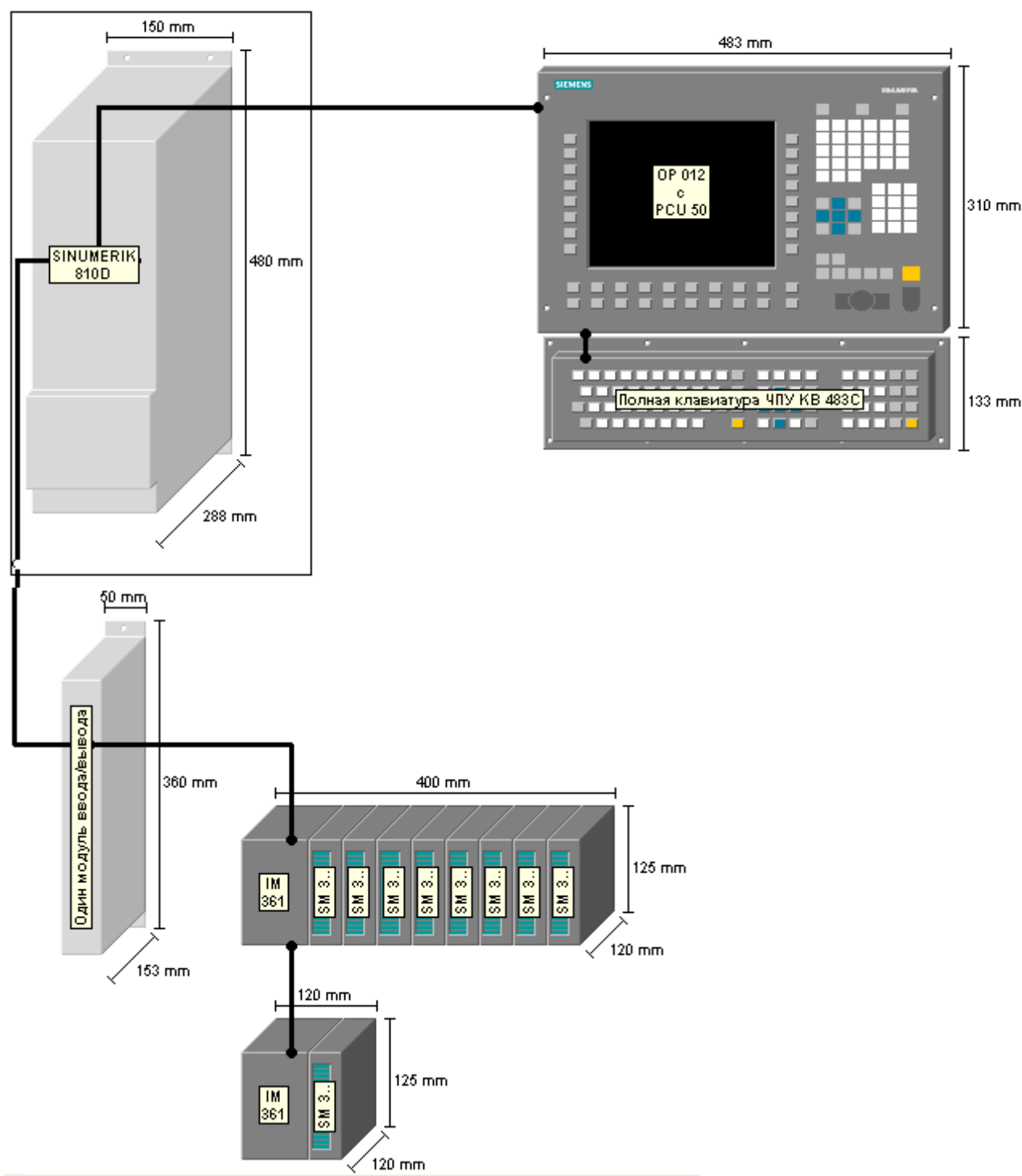


Рисунок 3.10 – Конструктивна вистава проекту системи ЧПУ

Конструктивне виконання системи керування дозволяє представити її компонування й розміщення. Так, наприклад, пристрій ЧПУ SINUMERIK 810D, змонтований в стійці перетворювача SIMODRIVE 611 (рис. 3.11),



модуль простої периферії EFP, а також контролер SIMATIC S7-300, що складається із двох стійок, з'єднаних один з одним інтерфейсними модулями ІМ 361, можуть бути розташовані в шафі керування. При цьому панель оператора із клавіатурою доцільно буде змонтувати безпосередньо на верстаті.

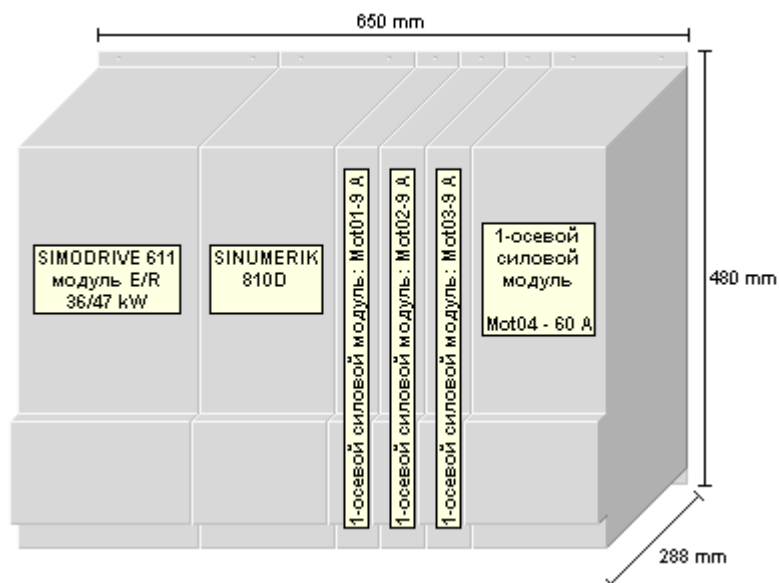


Рисунок 3.11 – Конструкція модулів SIMODRIVE 611

Програма NCSD дозволяє одержати відомості про потужність внутрішнього контуру, тобто розрахункової споживаної потужності верстата (рис. 3.12).

Мощность внутрен контура KONFIG~2.NC5					
Пожалуйста, выберите: Расчет для стандартного приложения					
двигатель	Тип двигателя	Стандартное приложение			
		Pcalc [кВт]	n/nII	PeI [кВт]	Сумма PeI [кВт]
Диапазон II для Pcalc от 1,8 кВт до 8,8 кВт					
Mot01	1FT6102-8AB71-1AG0	4,24	0,6	2,54	Сумма:
Mot02	1FT6102-8AB71-1AG0	4,24	0,6	2,54	
Mot03	1FT6102-8AB71-1AG0	4,24	0,6	2,54	
GLZ-Faktor (Standard): 0,5 x Сумма PeI + 10 % =				4,2	4,2
двигатель	Тип двигателя	РН [кВт]	КПД	PeI [кВт]	Допов. напр.:
Двигатели с рабочим циклом S1					
Mot04	1PH2113-6VWF42	15,1	0,8	18,9	25
GLZ-Faktor (Standard): 1 x Сумма PeI + 10 % =				20,8	

Рисунок 3.12 – Вивід сумарного значення споживаної потужності

Повний опис специфікації системи наведений у файлі Konfiguration\_Primer.xml, фрагмент якого представлено на рисунку 3.13.

```

<?xml version="1.0" encoding="UTF-8" ?>
- <ncsdkonf>
- <orderlist>
- <orderentry count="3" mlib="1FT6102-8AB71-1AG0" positionid="posid1">
- <description language="ru">
    SYNCHRONOUS SERVOMOTOR 1FT6
    <br />
    27NM, 100K,
    <br />
    1500RPM, 3,8KW
    <br />
    NATURAL AIR COOLING
    <br />
    IM B5 (IM V1, IM V3)
    <br />
    FOR POWER AND SIGNAL CONNECTOR
    <br />
    OUTGOING CONNECTOR DIRECTION
    <br />
    TRAVERSE RIGHT
    <br />
    INCREM. ENCODER, SIN/COS 1 VPP

```

Рисунок 3.13 – Фрагмент файлу специфікації, що описує двигун привода подачі

### 3.3 Індивідуальні завдання й вимоги до звіту по роботі

Індивідуальні завдання наведені в додатку Б. Номер варіанта визначається порядковим номером прізвища студента в журналі академічної групи.

Звіт по роботі повинен містити:

- завдання на конфігурування системи ЧПУ;
- конструктивне зображення системи;
- скриншот вікна “Потужність внутрішнього контуру”;
- роздрукований файл специфікації компонентів системи.

До звіту повинен додаватися файл конфігурації із розширенням .ncsd.

## 4 ПРОЕКТУВАННЯ ПРИВІДНОЇ СИСТЕМИ ВЕРСТАТА SINAMICS У СЕРЕДОВИЩІ ПРОГРАМИ SIZER

### 4.1 Розробка конструювання приводної системи верстата

Нехай у завданні на проектування визначені наступні характеристики приводної системи верстата:

1. Привод головного руху – потужність 25 кВт, номінальна частота обертання 1000 об/хв.
2. Приводи подач – 2 осі (X і Z), номінальний крутний момент 18 Нм, частота обертання 1500 об/хв ( по кожному приводу).
3. Пристрій керування – система ЧПУ Sinumerik 840Dsl із процесором NCU 710-2.
4. Панель оператора – OP 012 з PCU50.3.
5. Клавіатура PC – KB 310C.
6. Верстатний пульт – MCP 310C.
7. Блок живлення SITOP (24 В).

Запускаємо SIZER. Відкривається вікно програми з повідомленням, у якому вказується, що Siemens не відповідає за коректність і комплектність конфігурації, тобто відповідальність за це лежить на користувачі. Закриваємо це повідомлення й тиснемо на кнопку в лівому верхньому куті «New project».

Відкривається вікно New project (рис. 4.1), у якому вводимо ім'я проекту, наприклад, «Приводна система верстата».

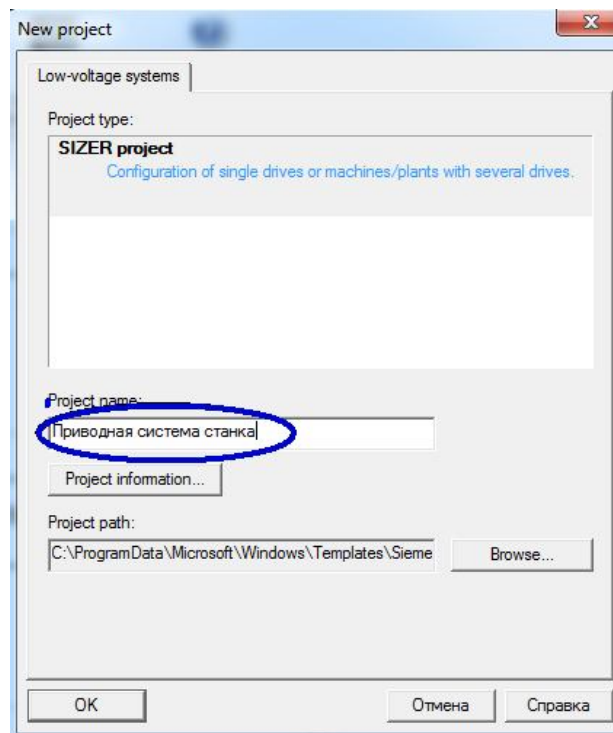


Рисунок 4.1

Закриваємо це вікно кнопкою ОК. Автоматично відкривається вкладка Drive systems (рис. 4.2) з іменем проекту й деревом завдань. Над деревом проекту перебуває меню Partial view для вибору виду вистави проекту.

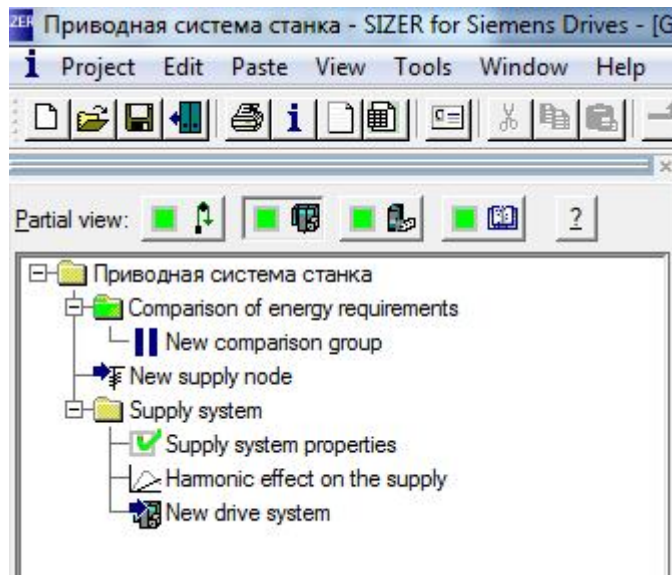


Рисунок 4.2

Кнопки цього меню дозволяють перемикатися між наступними вкладками:

- Mechanical Systems – механічна система;
- Drive systems – приводна система;
- Open-Loop/Closed-Loop Control / 24 V / Cabinet Module – вибір і настроювання засобів керування;
- Additional components – вибір додаткових компонентів.

Враховуючи те, що в завданні проектування приводної системи розробка механічних вузлів не передбачається, то залишаємося на вкладці Drive systems.

Дерево проекту на цій вкладці містить два розділи:

- Comparison of energy requirements – порівняння різних архітектур приводних систем (просто пропускаємо цей розділ).
- Supply system – система живлення.

Якщо подвійним клацанням миші відкрити Supply system properties, то можна побачити, що за замовчуванням призначена трифазна система живлення змінного струму напругою 400В частотою 50 Гц. Погоджуємося із цим варіантом.

Далі переходимо до Harmonic effect on the supply. При подвійнім клацанні лівою кнопкою відкривається вікно, у якому повідомляється, що для ряду систем, у тому числі й SINAMICS S120 гармонійний аналіз не проводиться, можливе тільки настроювання приводної системи. Закриваємо це вікно кнопкою ОК.

І, нарешті, переходимо до процесу компонування приводної системи. Двічі клацаємо лівою кнопкою по завданню New drive system. Відкривається вікно Add drive system, фрагмент якого показано на рисунку 4.3. Спочатку потрібно вибрати тип приводної системи – Single drives (одноосьовий привод), Multi-axis converter (багатоосевий привод) або Motor starter (нерегульований привод).

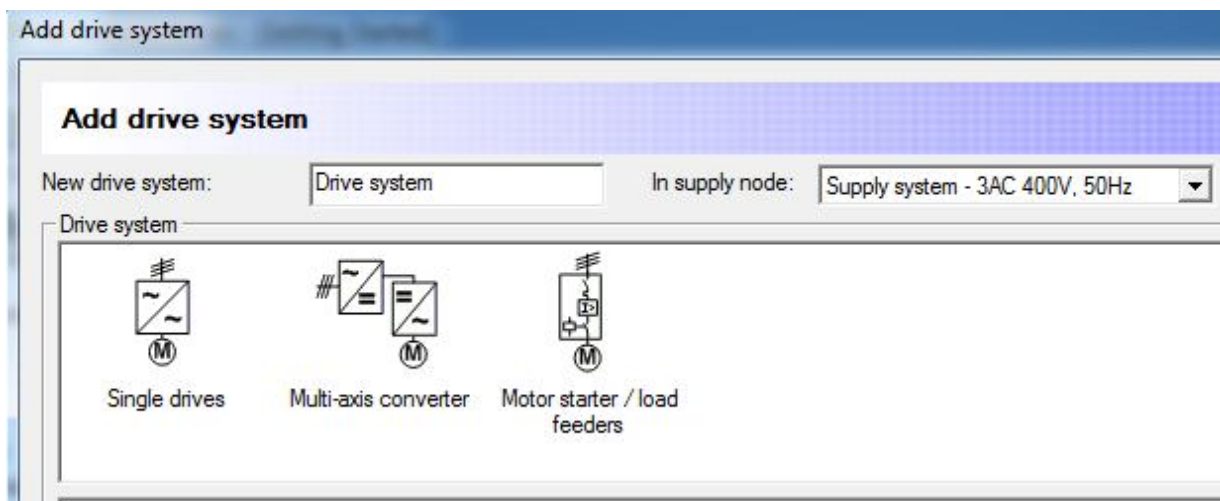


Рисунок 4.3

Вибираємо багатоосевий перетворювач (Multi-axis converter). При цьому праворуч від перетворювача стає доступним список для вибору версії компонування перетворювача. За замовчуванням показана версія Built-in unit version (інтегрований блок). Якщо відкрити список, то в ньому є можливість вибору Version in Cabinet Modules і S120 Combi system version.

Слід прийняти до уваги, що система Cabinet Modules з активним модулем живлення призначена для приводних систем від 132 кВт до 3500 кВт і виконується у форматі «шасі», а система Combi застосовується тільки в комбінації з перетворювачами типу Smart, тобто нерегульованими. Враховуючи те, що передбачається застосувати *регульований модуль* живлення, погоджуємося з версією Built-in unit, виділивши цей модуль, як показано на рисунку 4.4.

Далі переходимо до вибору модуля живлення. У якості модулів живлення перетворювача можуть використовуватися три модулі: Basic Line Module (без регулювання напруги проміжного контуру й без можливості рекуперації), Smart Line Module (без регулювання напруги проміжного контуру, але з можливістю рекуперації) і Active Line Module (з регулюванням напруги проміжного контуру й з можливістю рекуперації).

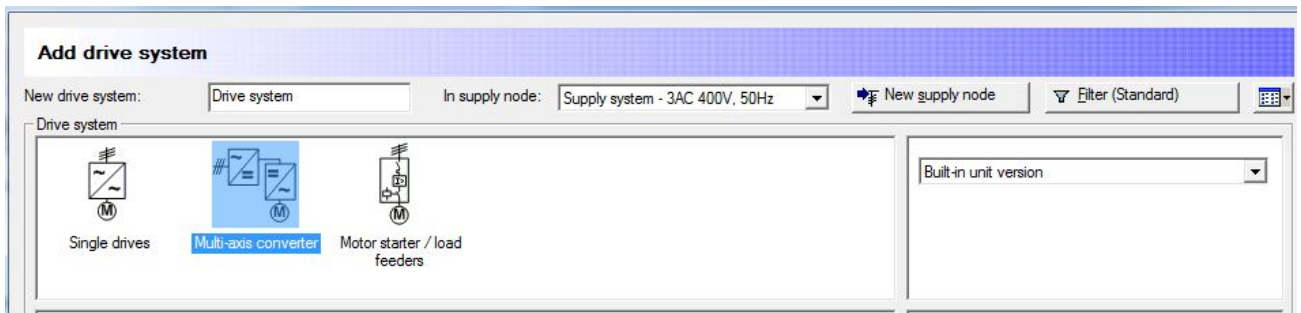


Рисунок 4.4

Враховуючи високі вимоги до динаміки приводів верстата, вибираємо активний модуль живлення (рис. 4.5). При виборі модуля праворуч стає доступним список варіантів охолодження модуля живлення. За замовчуванням список відкритий на варіанті *internal cooling* (внутрішнє охолодження). Погоджуємося із цим варіантом, зважаючи на те, що встановлена потужність приводів буде порівняно невеликою (близько 30 кВт).

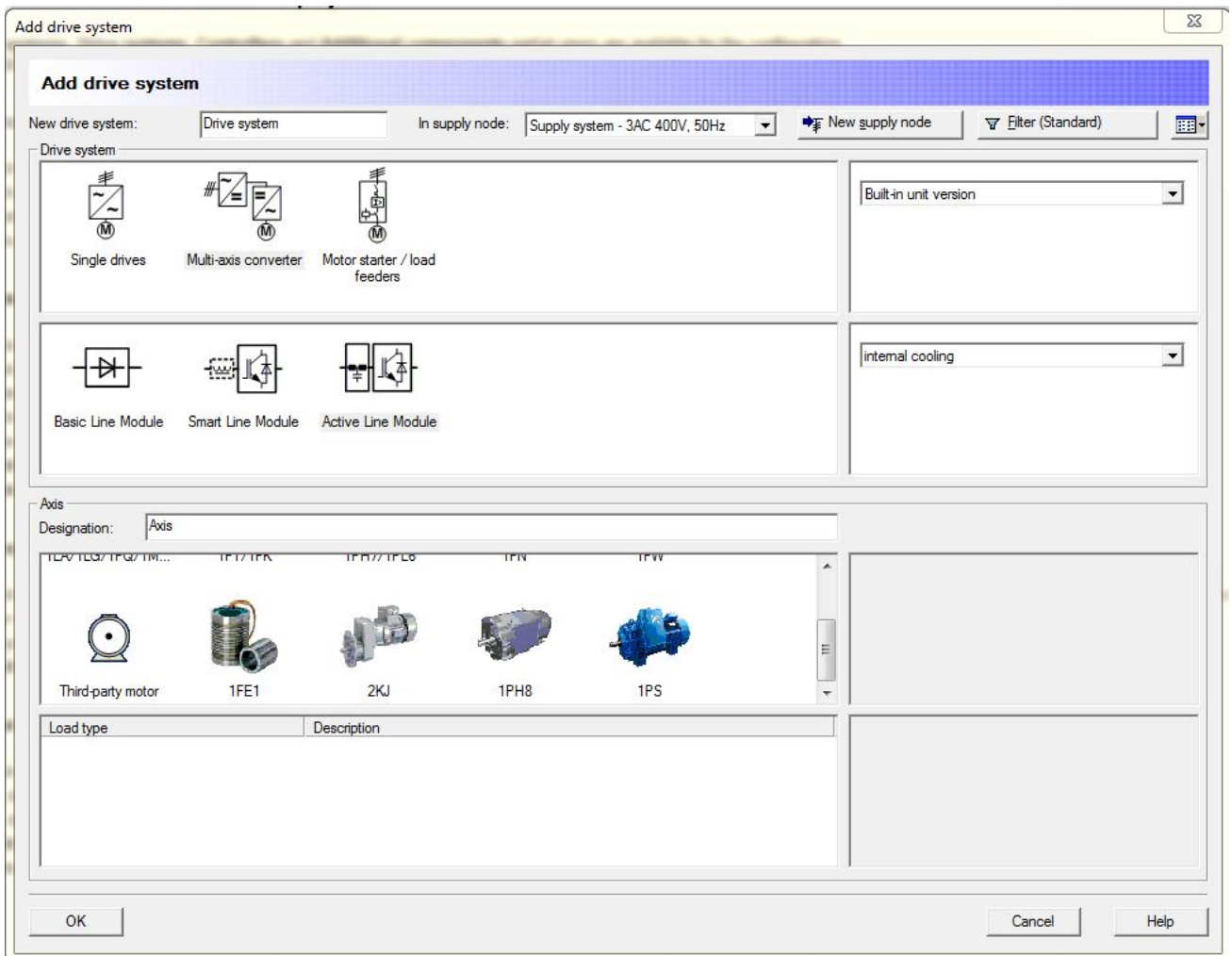


Рисунок 4.5



Далі в поле Designation (Позначення) уводимо назву першої осі – «Шпиндель» і вибираємо для цієї осі асинхронний двигун 1PH8. Праворуч від двигунів перебувають два списку, що розкриваються. Якщо розкрити верхній із цих двох списків, то можна побачити три варіанти режиму керування двигуном: Servo, Vector і V/f control. Вибираємо режим Servo і погоджуємося з тим, що інтегрованої функціональності привод не має (No drive-based integrated functionality). Після цих налаштувань вікно Add drive system здобуває вид, показаний на рисунку 4.6.

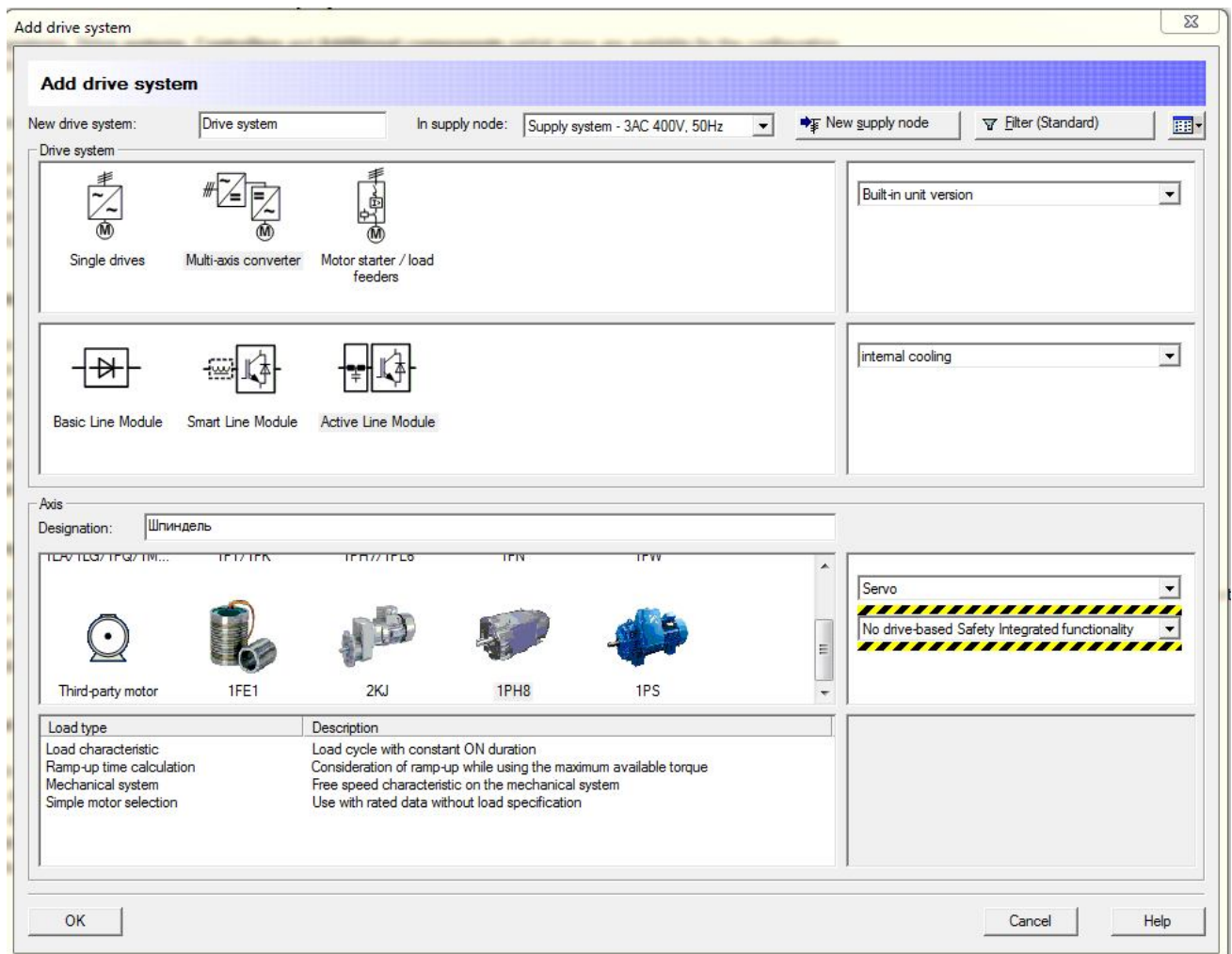


Рисунок 4.6

Слід прийняти до уваги, що програма відслідковує роботу з налаштуваннями, і якщо якийсь налаштування не виконувалося, то при спробі закрити вікно програма видає повідомлення: The selection is incomplete or the names are invalid, тобто «вибір є неповним». Тому слід пройти по всіх рядках у стовпці Description, клацнувши двічі на кожному з них.

При успішній завершенні налаштувань закриваємо вікно кнопкою ОК. При цьому в дереві проекту додається підрозділ «Шпиндель», у якому компоненти Motor, Motor Module і Output components ще не визначені (не відзначені зеленими галочками). Після закриття вікна Add drive system у робочій області (Workflow) виводиться панель графічного інтерфейсу з іконками Properties, Motor, Motor Module, Output components, Line Module і System components для керування процесом проектування.

Етапи проектування зв'язані стрілками, причому виконаний етап позначається зеленою галочкою. На рисунку 4.7 зеленою галочкою відзначена іконка Properties (Властивості).

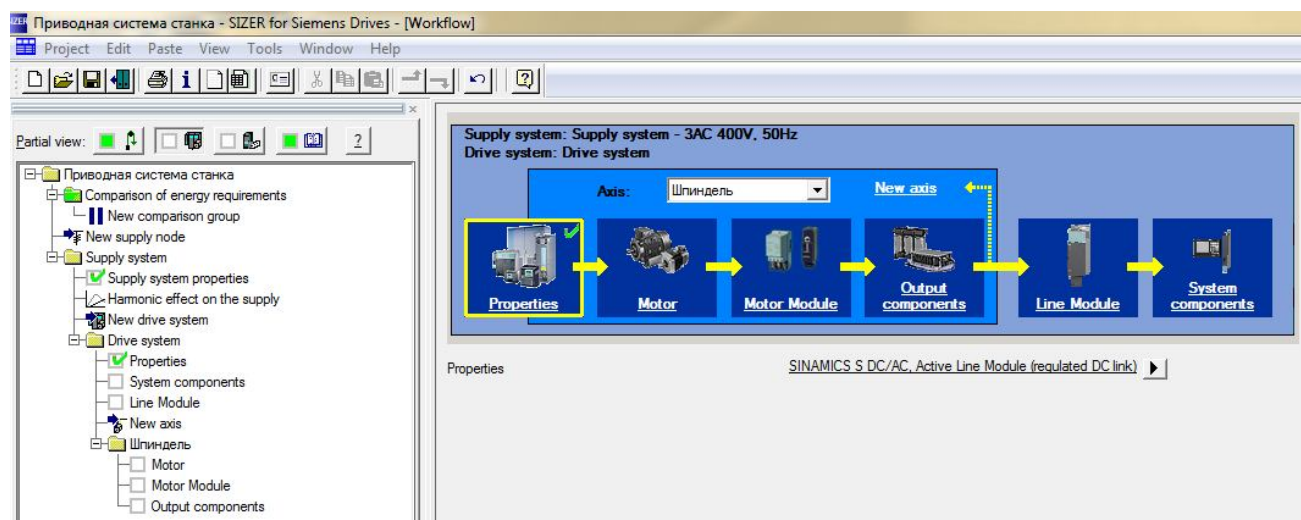


Рисунок 4.7

Продовжуємо компонувати приводну систему, додаючи в дерево проекту приводи подачі X і Z. Процедура для наступних приводів залишається тією ж, за винятком:

1. Перетворювач і модуль живлення не торкаємо, тому що вони загальні для всіх приводів.
2. У якості двигуна для приводів подачі використовуємо синхронні двигуни 1FT/1FK.

Двічі клацаємо по завданню New axis, знову відкривається вікно Add drive system, у якому записуємо *найменування осі X* у полі Designation і робимо вибір *типу двигуна*. Закінчивши налаштування для осі X, закриваємо вікно Add drive system кнопкою ОК і повторюємо всі процедури для осі Z. У результаті **дерево проекту** здобуває вид, показаний на рисунку 4.8.

Тепер можна переходити до конфігурування кожного привода. Порядок роботи із приводами не має значення, однак вибір або призначення параметрів по кожному з них проводиться в строгій послідовності.



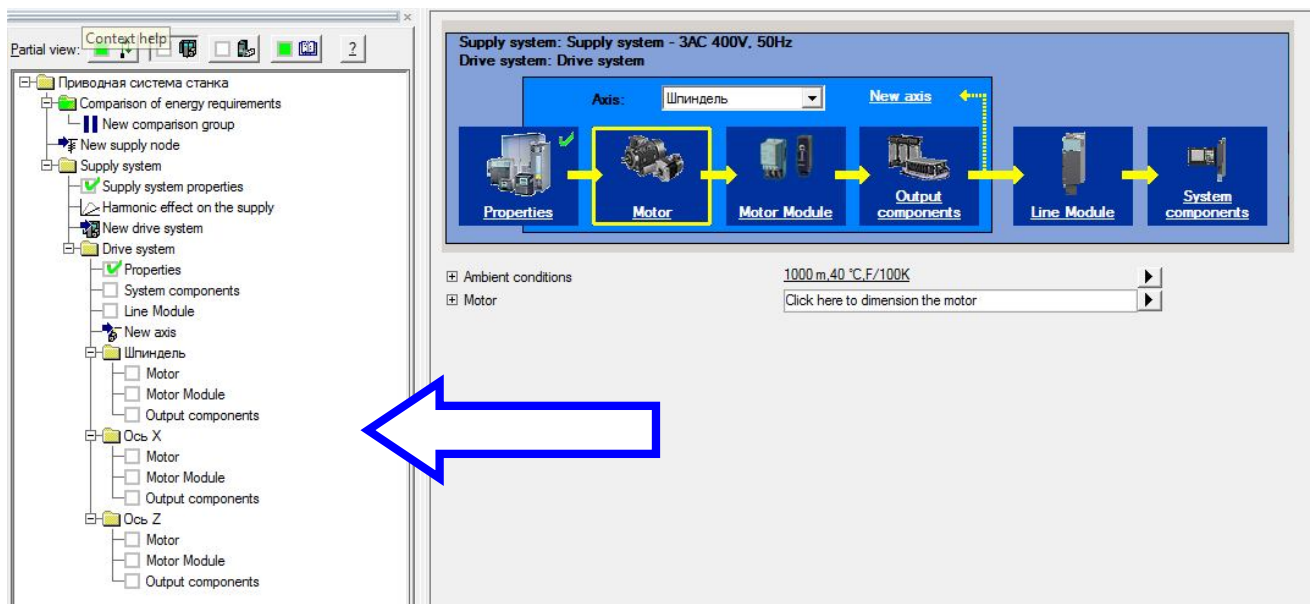


Рисунок 4.8

## 4.2 Конфігурування привода головного руху

Для проектування привода головного руху клацаємо двічі по завданню Motor у розділі «Шпиндель» дерева проекту. Однак зручніше користуватися панеллю графічного інтерфейсу з іконками Properties, Motor, Motor Module і т.д. Для цього вибираємо в списку, що розкривається, вісь Шпиндель і переконуємося, що іконка з написом Motor має жовту рамку, як показано на рисунку 4.9.

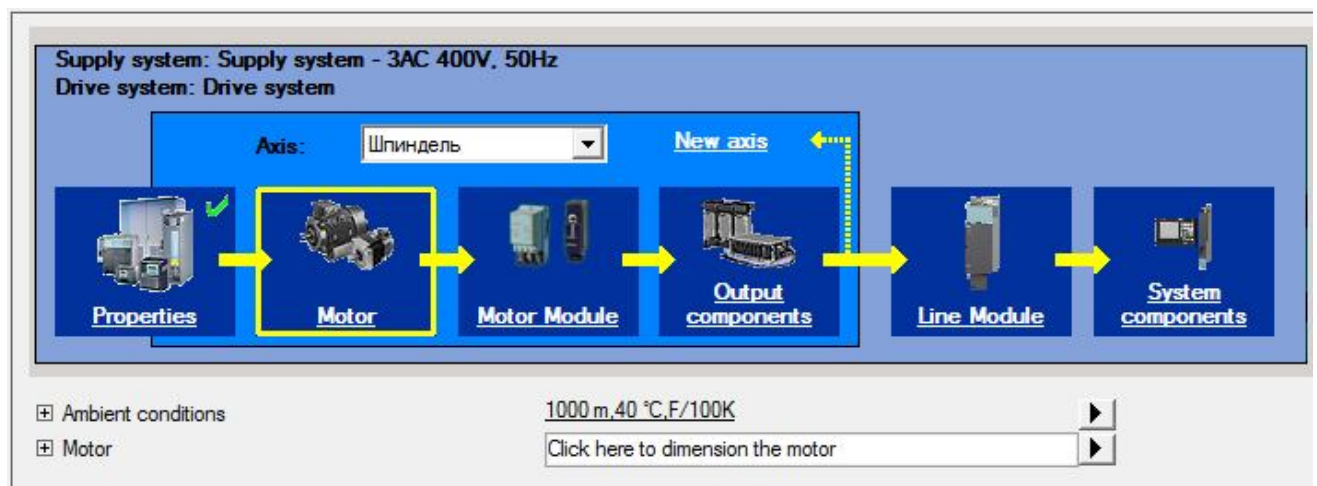


Рисунок 4.9

Під іконками графічного інтерфейсу розташовано два списки: Ambient conditions (Умови навколишнього середовища) і Motor (Двигун).

Умови навколишнього середовища можна змінити. Для цього клацаємо по кнопці ► наприкінці рядка й у вікні, що відкрилося, уводимо свої параметри. Однак, установлені за замовчуванням умови можна прийняти.

У рядку Motor додержуємося вказівки Click here to dimension the motor (Натисніть тут, щоб вибрати потужність мотора) і клацаємо по кнопці ►. При цьому відкривається вікно Motor wizard 1PH8 Step 1, яке показано на рисунку 4.10, у якому потрібно пройти чотири кроки настроювань. На першому кроці Basic data 1 необхідно ввести базові дані.

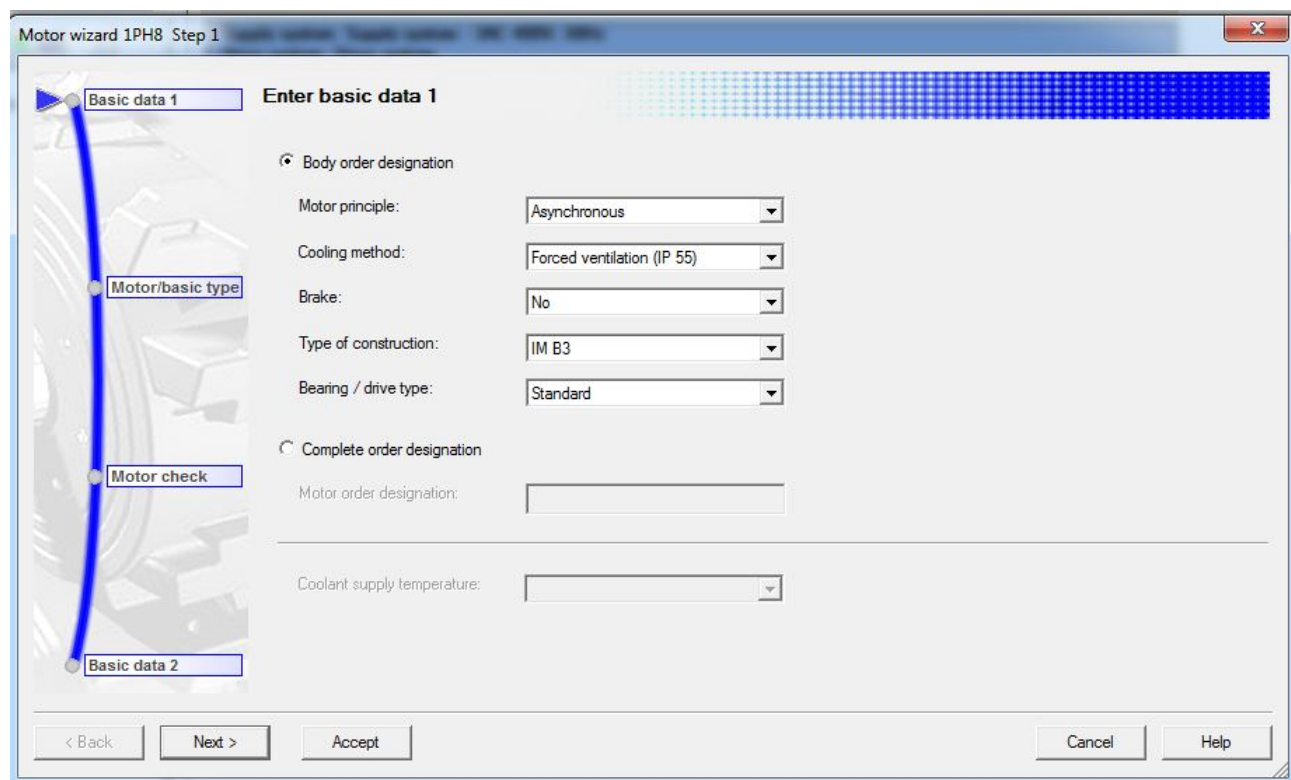


Рисунок 4.10

Насамперед, слід визначитися з методикою (порядком) вибору – чи то призначаємо параметри послідовно, тоді включаємо прапорець на Body order designation, чи то вказуємо конкретний замовлений номер, тоді включаємо прапорець на Complete order designation (Повна вказівка замовлення).

У наших умовах застосовуємо варіант послідовного вибору, тому включаємо кнопку на Body order designation.

Далі послідовно (зверху вниз) вибираємо: асинхронний двигун, охолодження примусовою вентиляцією (виконання IP55), без застосування гальма (Brake – No), застосовуємо конструкцію типу IM B3 (горизонтальна вісь обертання), застосовуємо підшипники (Bearing) стандартного типу.

Після цього натискаємо кнопку Next і переходимо до наступного кроку – вибору конкретного типу двигуна. У вікні, показаному на рисунку 4.11, представлені дані двигунів потужністю від 3 кВт до 475 кВт.

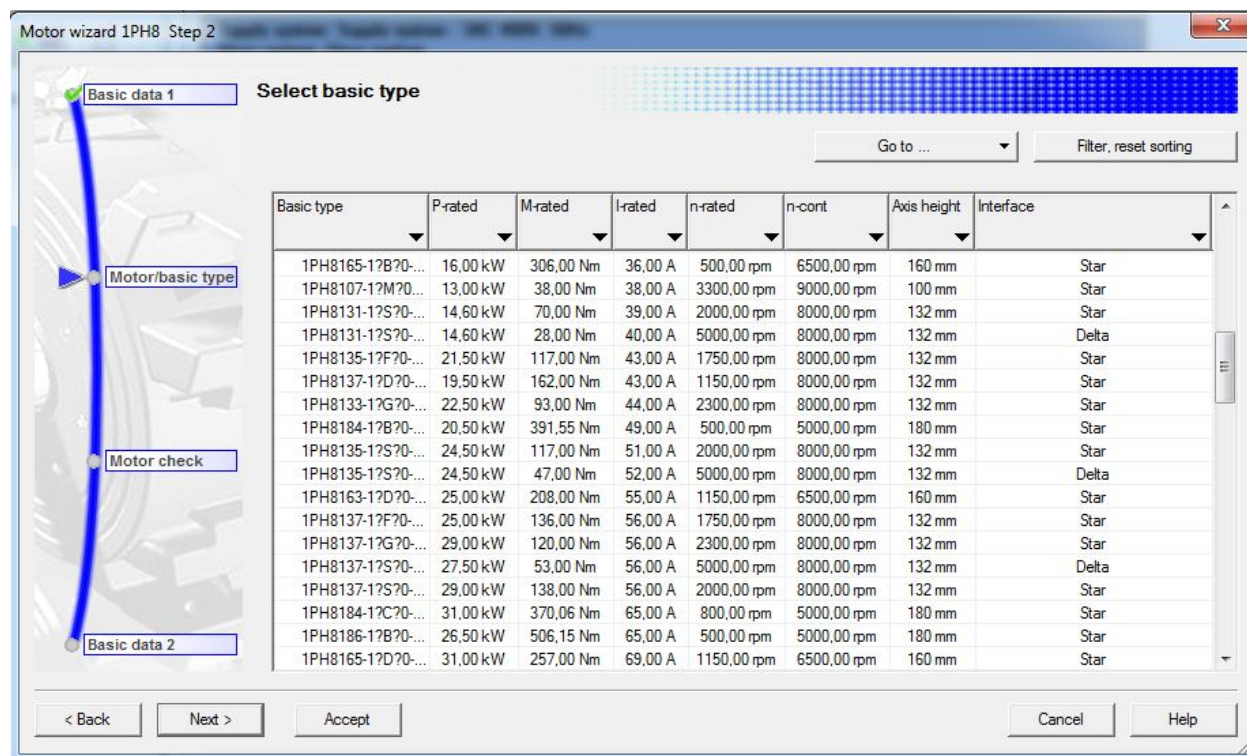


Рисунок 4.11

Щоб спростити процес вибору двигуна у великому списку, можна включити фільтри, які настроюються в списках стовпців.

Для цього в стовпці **P-rated** клацаєм по кнопці ▼ і для вибору двигуна потужністю 25 кВт у списку, що відкрився, призначаємо діапазон 10-110 кВт, а в **стовпці n-rated** для вибору двигуна зі швидкістю 1000 об/хв призначаємо діапазон 750-1500 об/хв. У результаті список типів двигунів скорочується до 13 позицій (рис. 4.12), з них тільки один тип задовольняє вимогам завдання найбільшою мірою – двигун 1PH8163 потужністю 25 кВт і номінальною швидкістю обертання 1150 об/хв.

Виділяємо обраний двигун і клацаємо по кнопці Next. Відкривається наступне вікно Motor check, показане на рисунку 4.13, у якому виводяться робочі характеристики двигуна.

Закриваємо вікно Motor check кнопкою Next і переходимо до останнього кроку – Basic data 2, вид вікна якого показано на рисунку 4.14.

У цьому вікні слід відкрити список Motor encoder і вказати чи буде двигун з датчиком або без датчика (Without encoder). Якщо двигун повинен бути з датчиком (для верстатних приводів датчик обов'язковий), то в списку вибираємо один з типів датчиків. Для того, щоб контролювати кут повороту



ротора, слід застосовувати інкрементальні датчики з додатковими C і D доріжками. Тому вибираємо: «Incremental encoder sin/cos 1 Vpp 2048 S/R with C, D track».

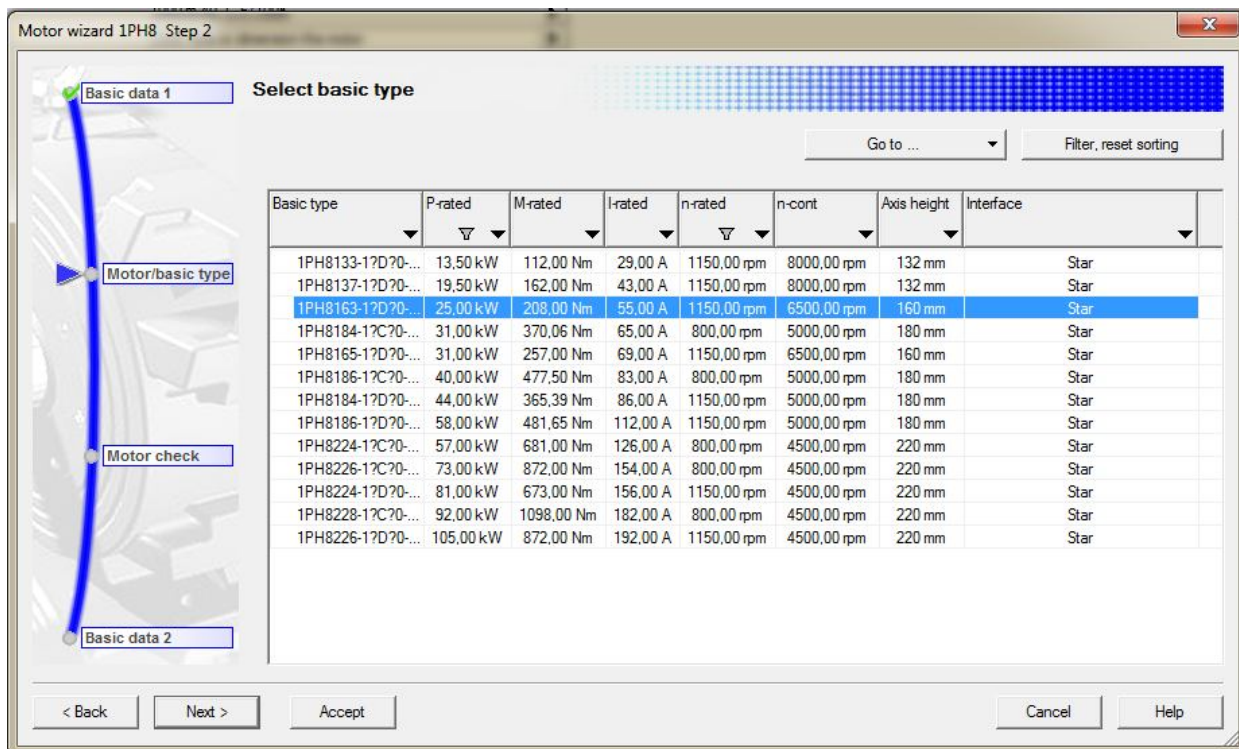


Рисунок 4.12

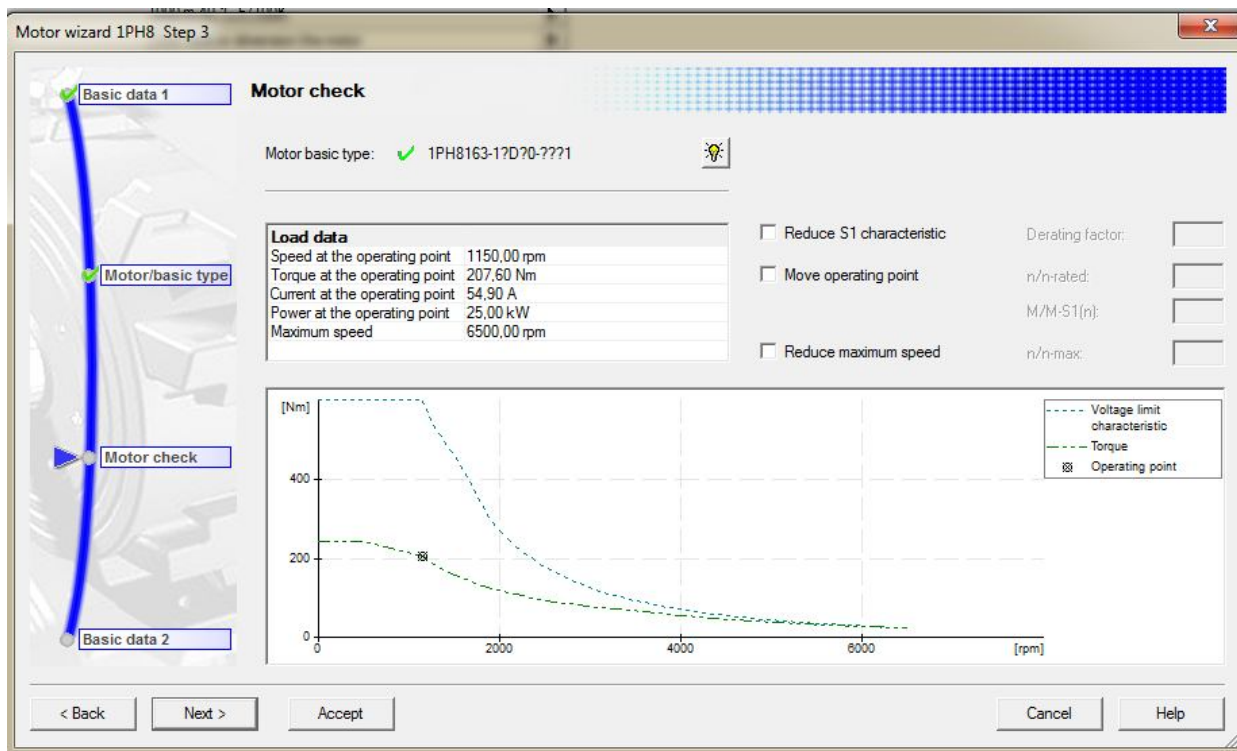


Рисунок 4.13

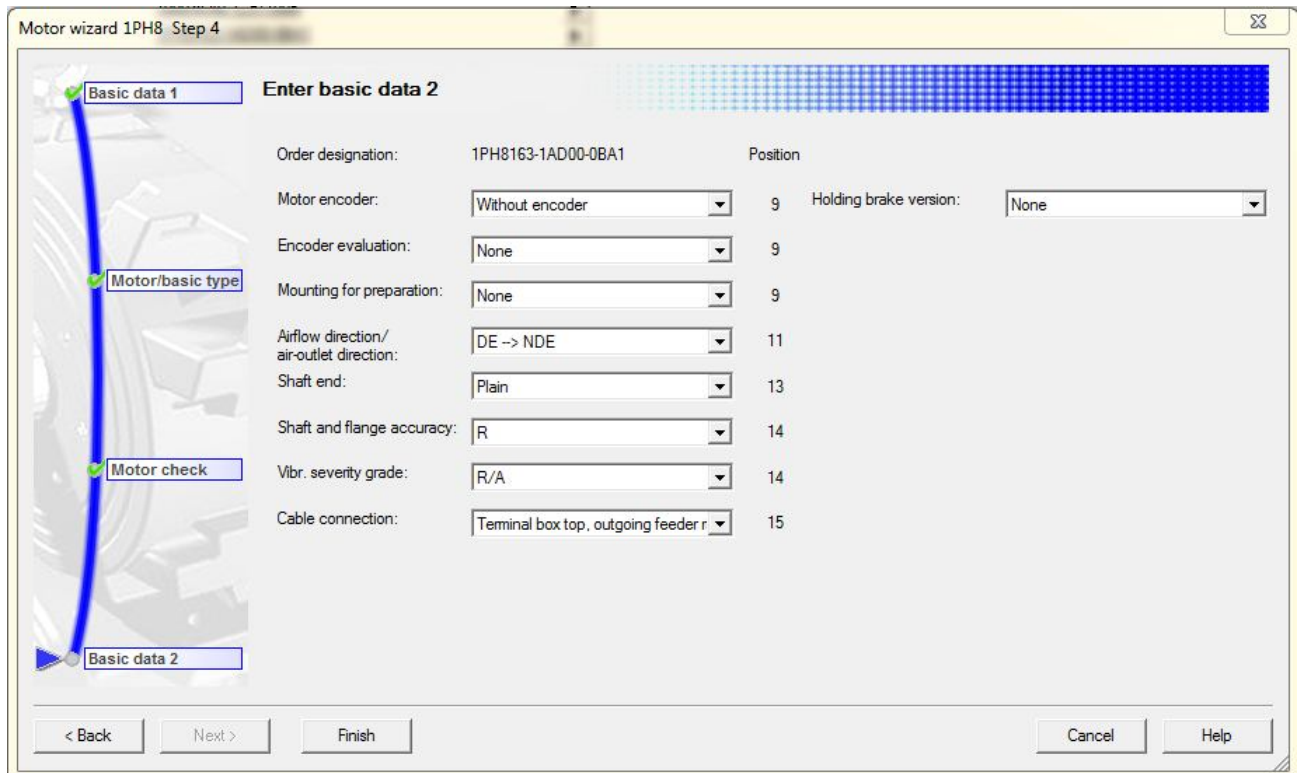


Рисунок 4.14

У наступному рядку – Encoder evaluation (Обробка сигналів датчика) – програма пропонує єдиний розв'язок: установити зовнішній SMC модуль датчика для переходу від сигналів sin/cos 1 Vpp до інтерфейсу DRIVE-CLIQ.

У рядку Mounting for preparation (Кріплення) вибір відсутній.

У рядку Airflow direction (Напрямок потоку повітря) вибираємо варіант DE→NDE. Потік повітря стандартно спрямований від вала двигуна (DE) до задньої сторони двигуна (NDE) для відводу втрат тепла двигуна від верстата. Далі погоджуємося з усіма рекомендаціями, що стосуються кінця вала, фланця, рівня вібрацій і напрямку виводу кабелю.

Натискаємо кнопку Finish і переконуємося, що в графічному інтерфейсі Workflow на іконці Motor для осі Шпindel з'явилася галочка завершення процедури.

Одночасно під графічним інтерфейсом додалися дві нові кнопки зі списками, що розкриваються, – Ramp-up time calculation (Розрахунки часу розгону) і Options (Опції). Перша кнопка відкриває вікно введення параметрів для розрахунків часу розгону двигуна, друга кнопка дозволяє вказати опції, тобто необов'язкові компоненти привода. Пропускаємо ці операції й тиснемо на іконку Motor Module осі Шпindel.

В оновленому інтерфейсі (рис. 4.15) іконка Motor Module забезпечується жовтою рамкою. Нижче розташовано два списки – Ambient conditions (Умови

нарколишнього середовища) і Motor Module. Клацаємо по кнопці ► у рядку Motor Module. Відкривається вікно для вибору модуля двигуна (рис. 4.16).

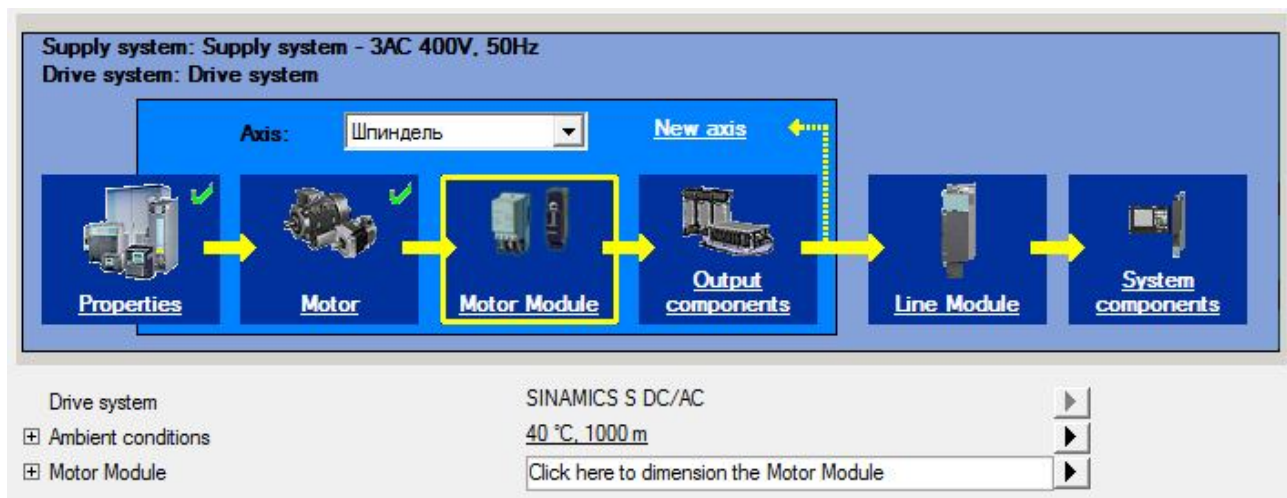


Рисунок 4.15

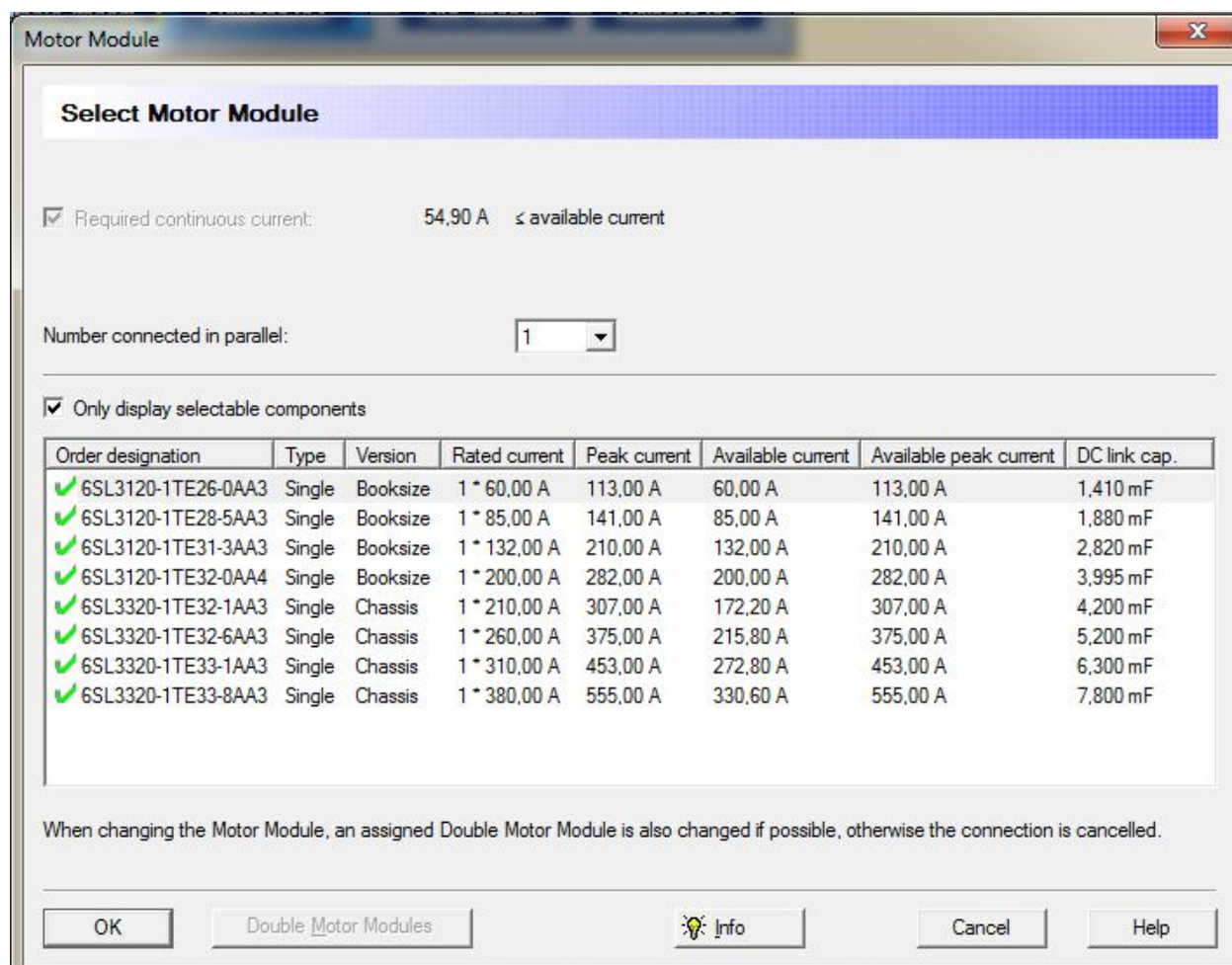


Рисунок 4.16



У вікні Motor Module виведений список припустимих по струму варіантів модулів двигунів, а сірим кольором підфарбований найбільш підходящий варіант. Вибираємо цей варіант і закриваємо вікно кнопкою ОК. При цьому в рядку Motor Module з'являється замовлений номер модуля, а нижче стає доступним новий рядок – Accessories (Акcesуари).

Враховуючи, що додаткових приналежностей не потрібно, переходимо до наступного етапу й клацаємо по іконці Output components.

На графічному інтерфейсі (рис. 4.17) іконка Output components здобуває жовту рамку (активна фаза), а завершені (попередні) етапи позначаються зеленими галочками, що означає «так; перевірене».

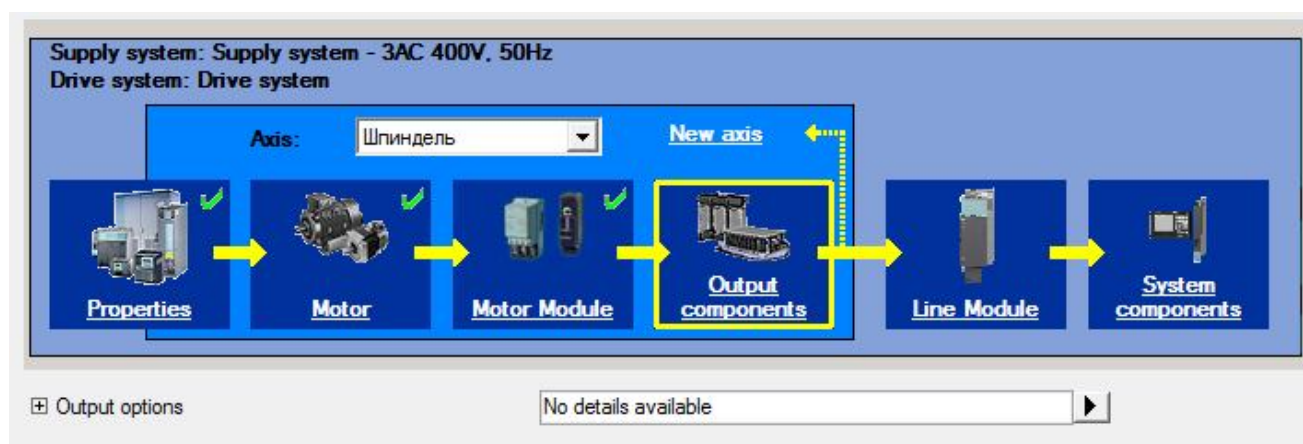


Рисунок 4.17

У рядку Output options у текстовім полі написано: No details available (Немає доступних подробиць). Клацаємо по кнопці ►, відкривається вікно Output options, яке слід просто закрити через відсутність доступних операцій.

Однак після закриття вікна Output options під іконками стають доступні два нові списки (рис. 4.18) – Supply cable (Кабель живлення) і Encoder evaluation (Визначення енкодера).

Натискаємо кнопку ► на рядку Supply cable. При цьому відкривається однойменне вікно (рис. 4.19), у якому тип кабелю й перетин жил уже визначені. Залишається тільки ввести в поле введення стовпця Length (Довжина) необхідну довжину кабелю. Враховуючи, що конкретні дані по монтажу в нас відсутні, погоджуємося із запропонованою в програмі довжиною 10 м і закриваємо вікно.

Переходимо до рядка Encoder evaluation і натискаємо кнопку ►, при цьому відкривається однойменне вікно, фрагмент якого показано на рисунку 4.20.

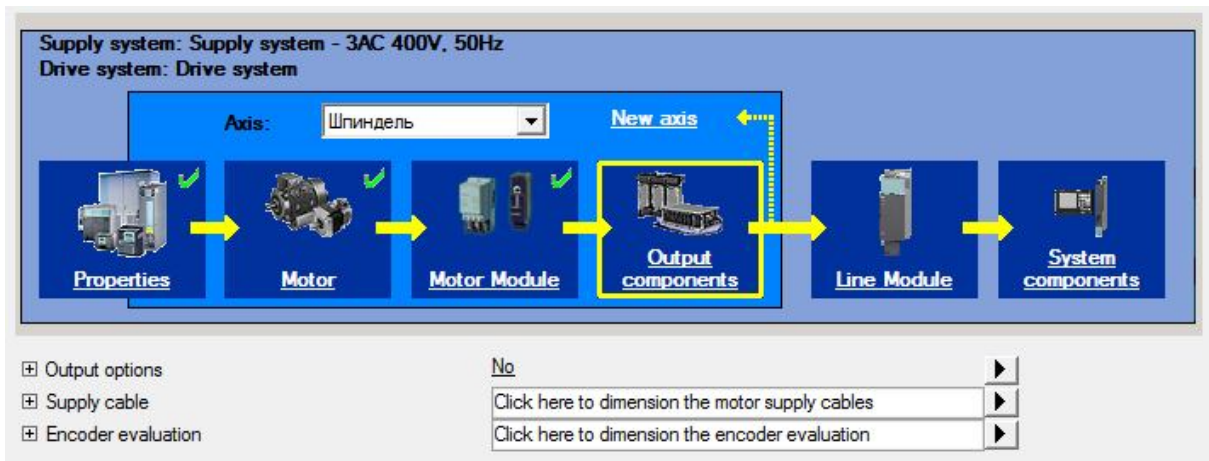


Рисунок 4.18

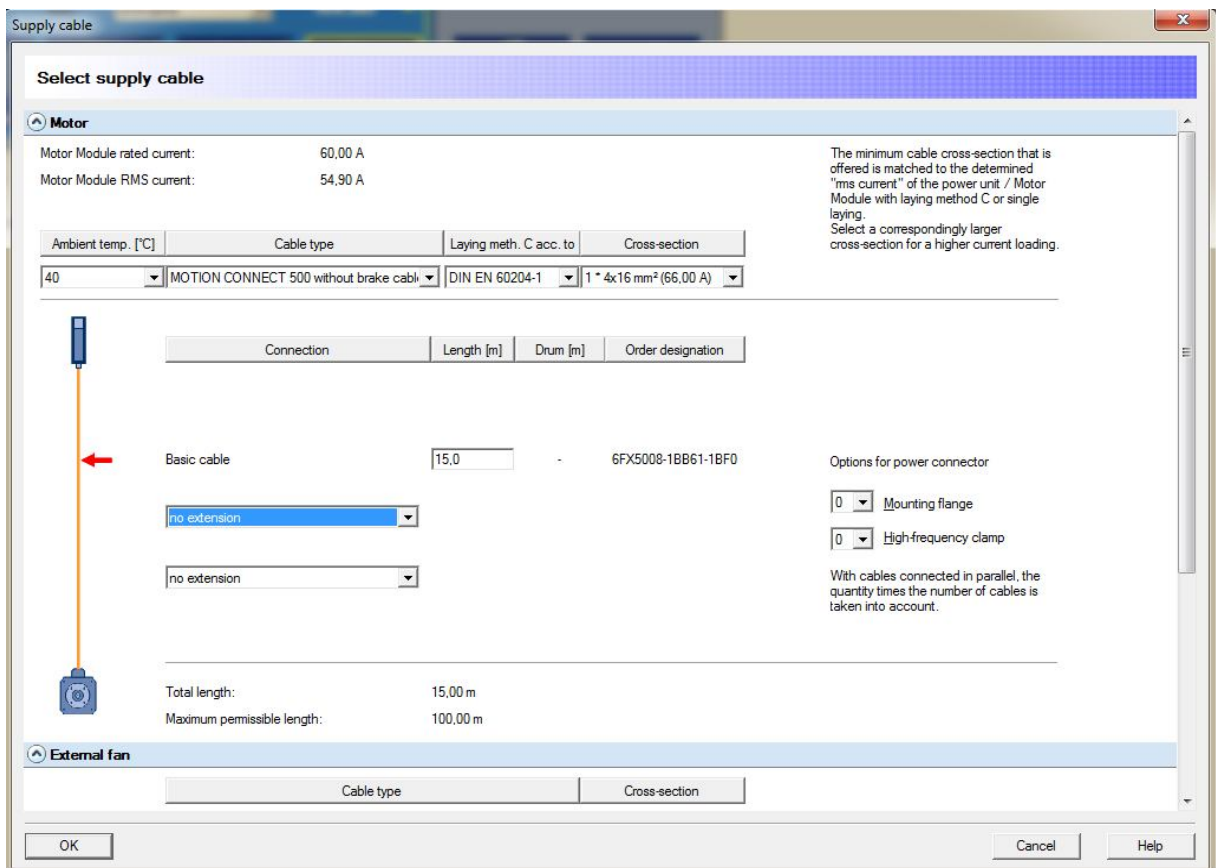


Рисунок 4.19

У цьому вікні вгорі відображається обраний раніше енкодер (вбудований у двигун), а внизу можна додати зовнішній енкодер для системи прямого контролю положення робочого органа. Вікно можна закрити кнопкою ОК як із призначенням кабелю для з'єднання енкодера з модулем керування, так і без призначення.

Закриваємо вікно Encoder evaluation, завершуючи тим самим проектування осі «Шпиндель».



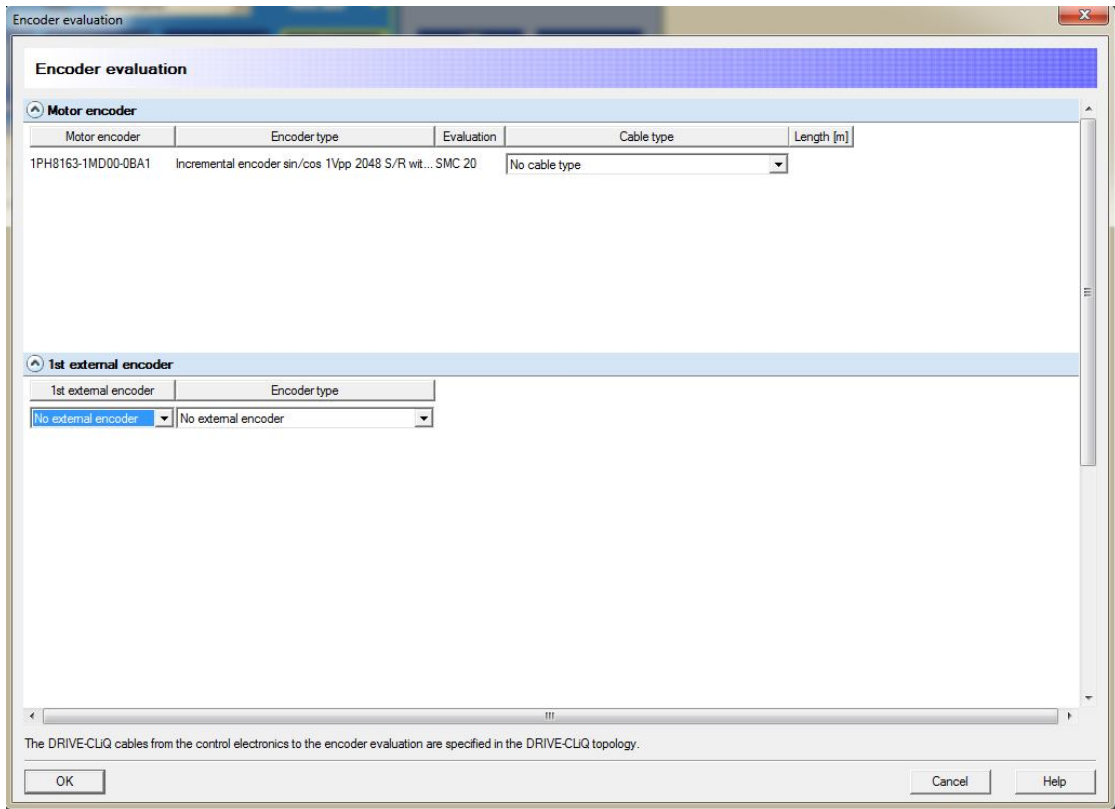


Рисунок 4.20

Пунктирна лінія на графічному інтерфейсі (рис. 4.21) показує, що потрібно змінити вісь і повторити для нової осі етапи Motor, Motor Module і Output components. Тільки закінчивши конфігурування приводів усіх осей, можна буде переходити до проектування Line Module і System components.

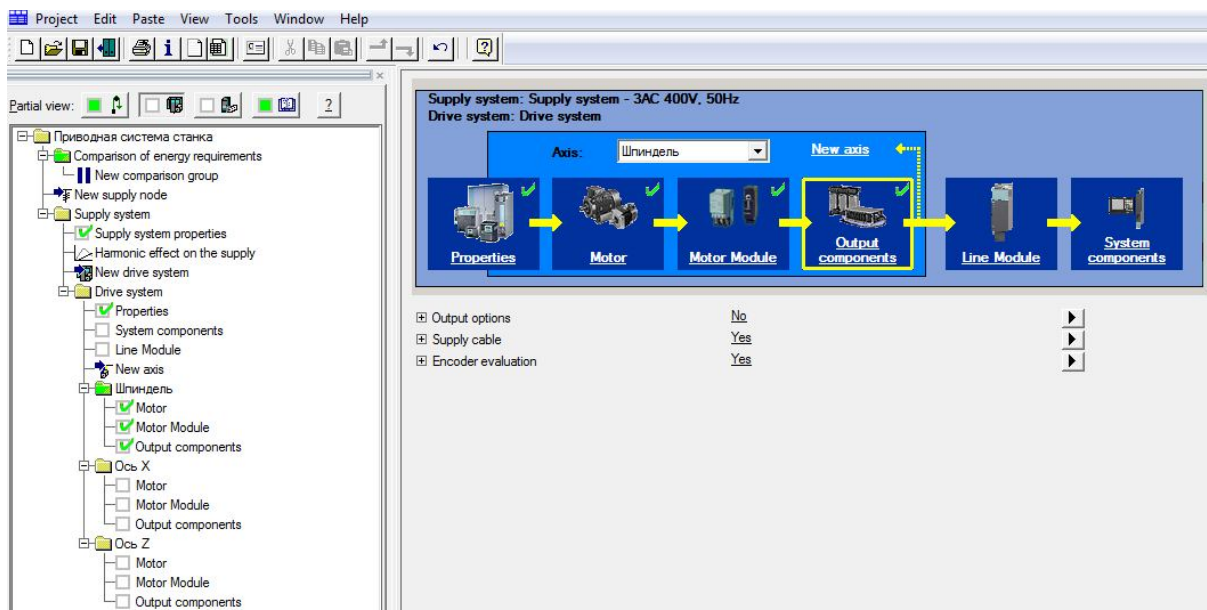


Рисунок 4.21

### 4.3 Конфігурування приводів подач

Переходимо до проектування привода подачі осі X. Для цього в списку Axis вибираємо вісь X і клацаємо по іконці Motor. У результаті з'являються два списки – Ambient conditions (Умови навколишнього середовища) і Motor (Двигун), як показано на рисунку 4.22.

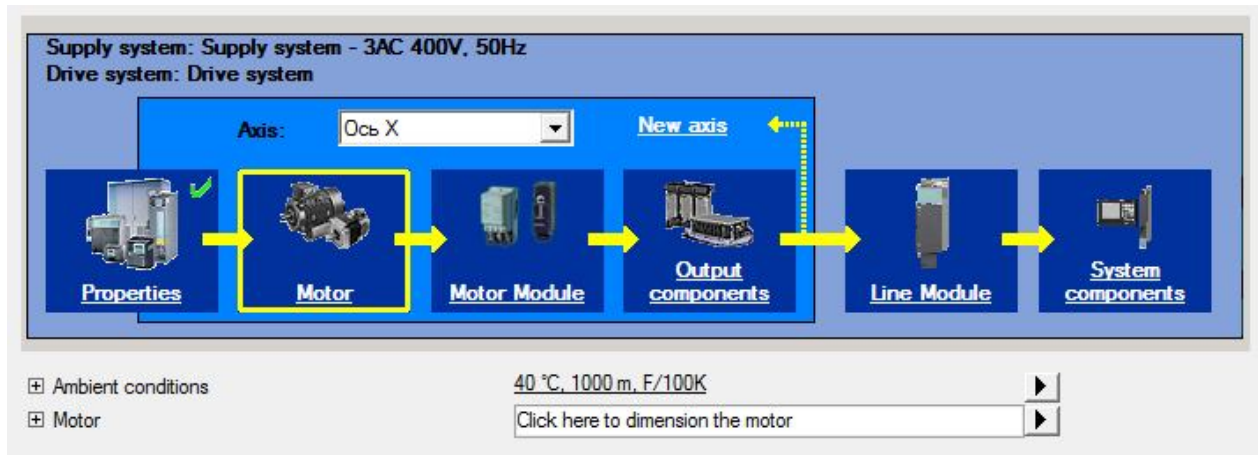


Рисунок 4.22

Умови навколишнього середовища залишаємо без зміни.

У рядку Motor додержуємося вказівки Click here to dimension the motor (Натисніть тут, щоб вибрати потужність мотора) і клацаємо по кнопці ►. При цьому відкривається вікно Motor wizard 1FT/1FK Step 1, показане на рисунку 4.23, у якому необхідно пройти п'ять кроків налаштувань.

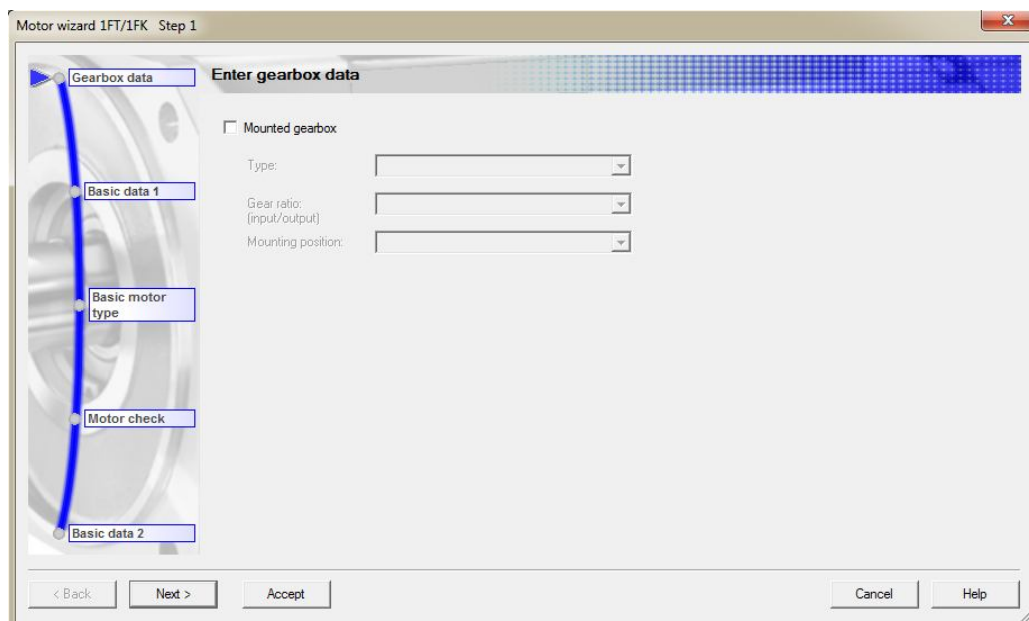


Рисунок 4.23

На першому кроці необхідно ввести базові дані по редуктору (Gearbox data), якщо він повинен бути скомпонований із двигуном. Як приклад на рисунку 4.24 показаний загальний вид двигуна 1FT6 із прибудованим планетарним редуктором типу SP+.



Рисунок 4.24

Враховуючи те, що застосування двигуна з редуктором не передбачене, пропускаємо цей крок і, натиснувши кнопку Next, переходимо до наступного кроку – Basic data 1, інтерфейс якого представлено на рисунку 4.25.

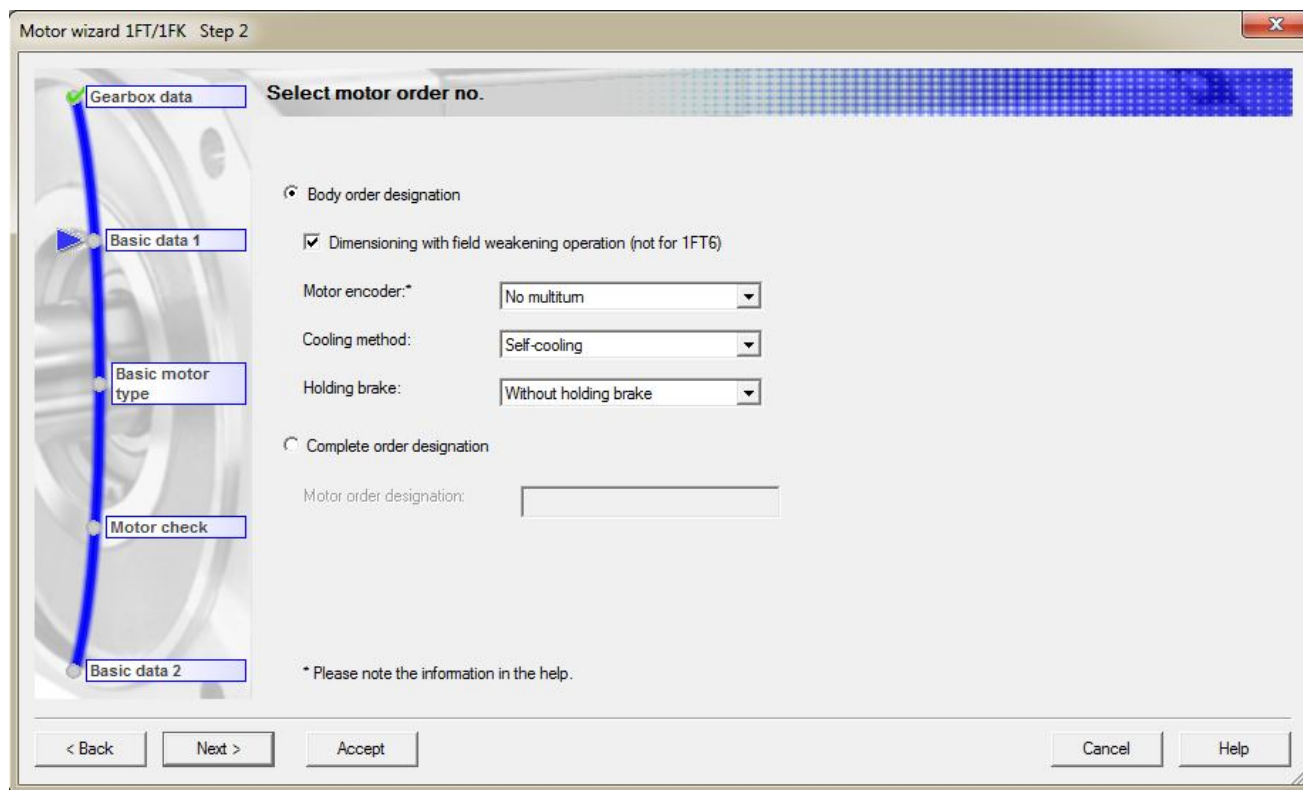


Рисунок 4.25

На цьому кроці слід включити кнопку *Body order designation* (Послідовне конфігурування привода), а також зняти прапорець на рядку «*Dimensioning with field weakening operation*» (Установка параметрів з урахуванням режиму ослаблення поля). Якщо прапорець не зняти, то обумовлена в програмі потужність двигуна для роботи в умовах ослаблення поля буде значно вище. Далі відмовляємося від багатооборотного енодера (*No multitum*) у двигуні, вибираємо спосіб охолодження – *self cooling* (самоохолодження) і, нарешті, відмовляємося від стояночного гальма (*without holding brake*).

Натиснувши кнопку *Next*, переходимо до кроку 3 – *Basic motor type*, інтерфейс якого показано на рисунку 4.26. Тут показаний результат вибору двигуна, отриманий при настроюванні фільтрів **M-rated** на діапазон 10-25 Нм (завдання  $M=18$  Нм) і **n-rated** на 1500 об/хв (завдання  $n=1500$  об/хв). При цьому отриманий єдиний результат. Відзначаємо його клацанням лівої кнопки й тиснемо *Next* для переходу до наступного кроку *Motor check*.

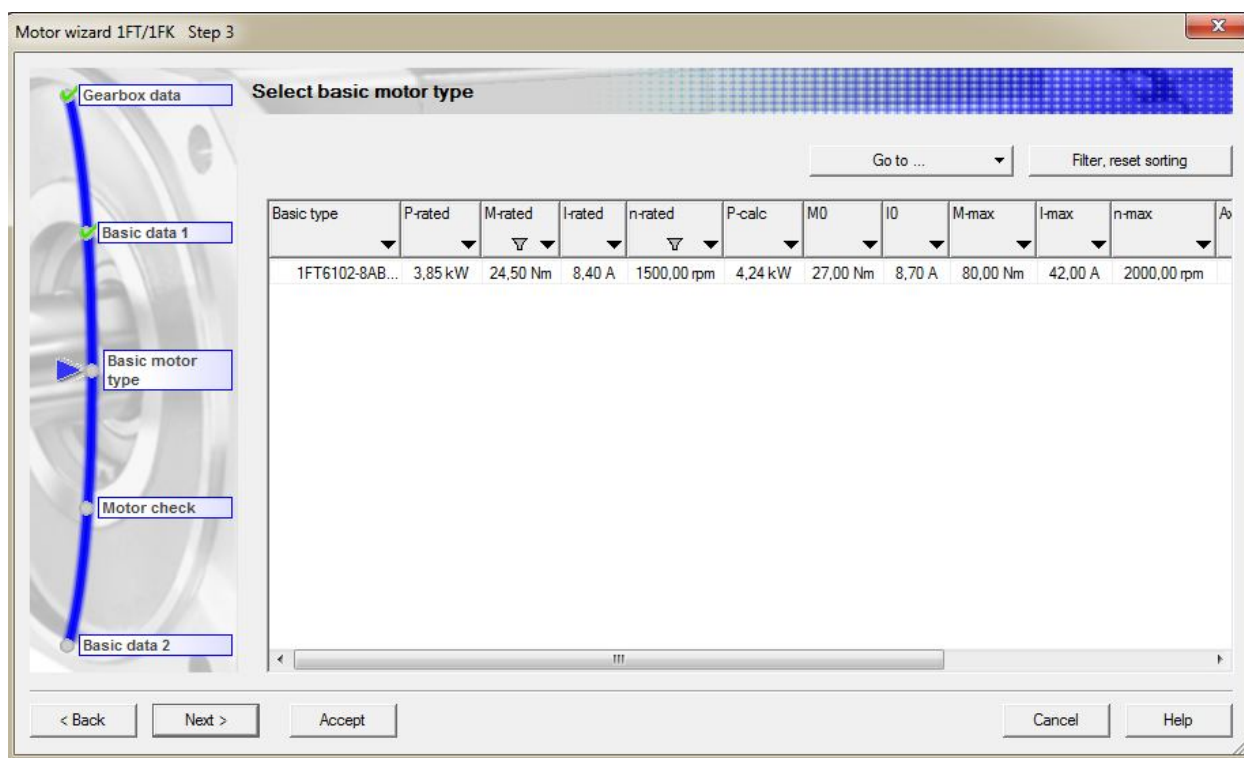


Рисунок 4.26

У вікні *Motor check* (рис. 4.27) представлені результати розрахунків робочих характеристик на номінальній швидкості. Вони свідчать про те, що в обраного двигуна крутний момент на номінальній швидкості перевищує задане значення, а значить задовольняє вимогам.

Приймаємо це до відома й переходимо до останнього кроку *Basic data 2*. На рисунку 4.28 показаний вид вікна *Basic data 2*.

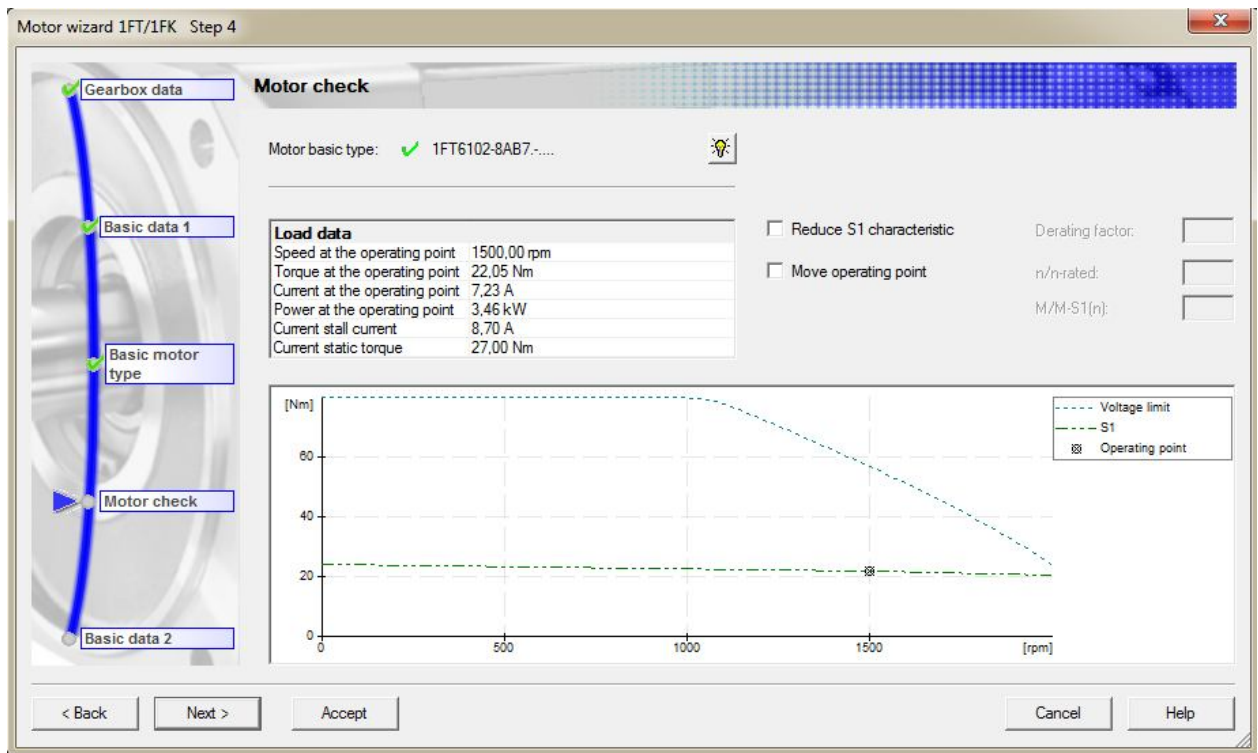


Рисунок 4.27

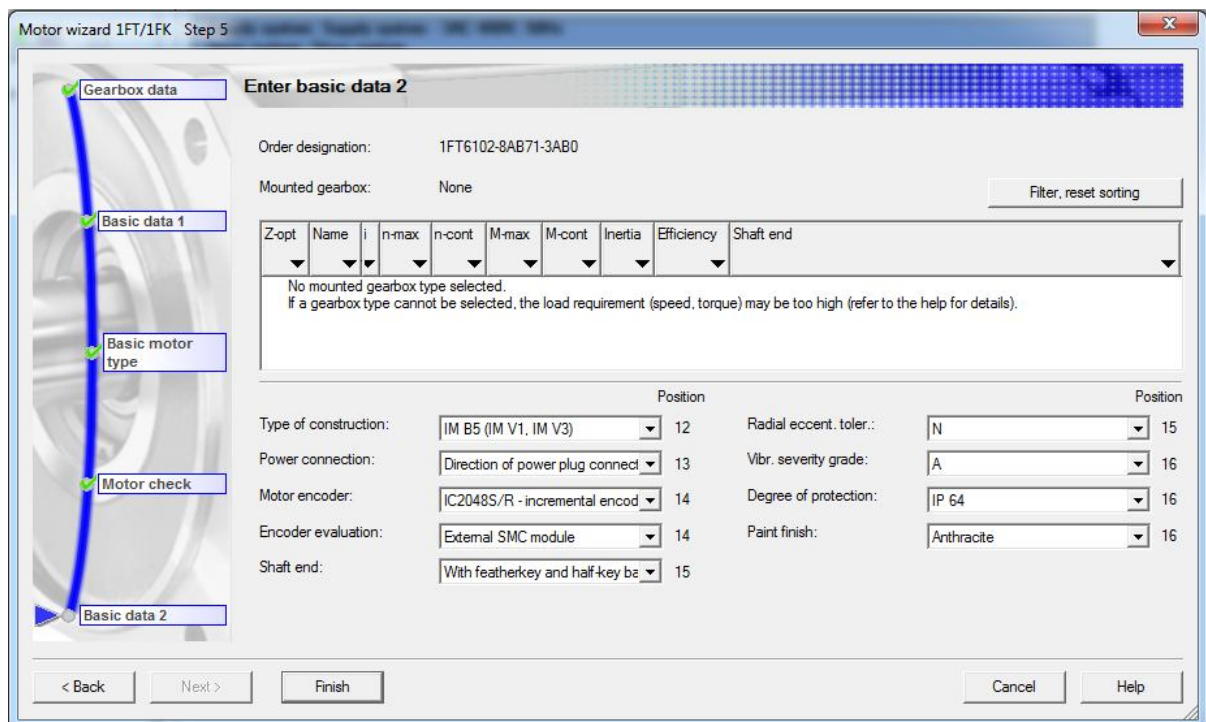


Рисунок 4.28

Прийнявши до уваги, що двигун не має вбудованого редуктора, приходимо до висновку, що верхня частина цього вікна не несе ніякої інформації.



Починаючи з позиції 12 послідовно призначимо найбільш важливі дані. У позиції 12 тип конструкції визначається розташуванням вала. Якщо після IM стоїть буква B, то це горизонтальне розташування вала двигуна, якщо буква V, то вал розташований вертикально. Ухвалюємо горизонтальне розташування й у позиції 12 встановлюємо IM B5.

Далі приймаємо вказаний напрямок виходу силового кабелю в позиції 13. У позиції 14 для обраного двигуна 1FT6 вибираємо інкрементальний енкодер з доріжками C і D без інтерфейсу DRIVE-CLIQ. У позиції 15 погоджуємося із застосуванням зовнішнього SMC модуля для сполучення енкодера з DRIVE-CLIQ інтерфейсом. У наступних позиціях слід погодитися з усіма пропозиціями й закінчити роботу із двигуном, натиснувши кнопку Finish.

Після закриття вікна Motor wizard на іконці Motor осі X з'являється зелена галочка, що свідчить про успішне закінчення цієї фази.

Клацаємо по іконці Motor Module, а потім на рядку Motor Module по кнопці ►. У вікні, що відкрилося, яке показано на рисунку 4.29, виведений список модулів, які задовольняють вимогам по навантаженню, при цьому перший модуль у списку підфарбований сірим кольором, як найбільш підходящий.

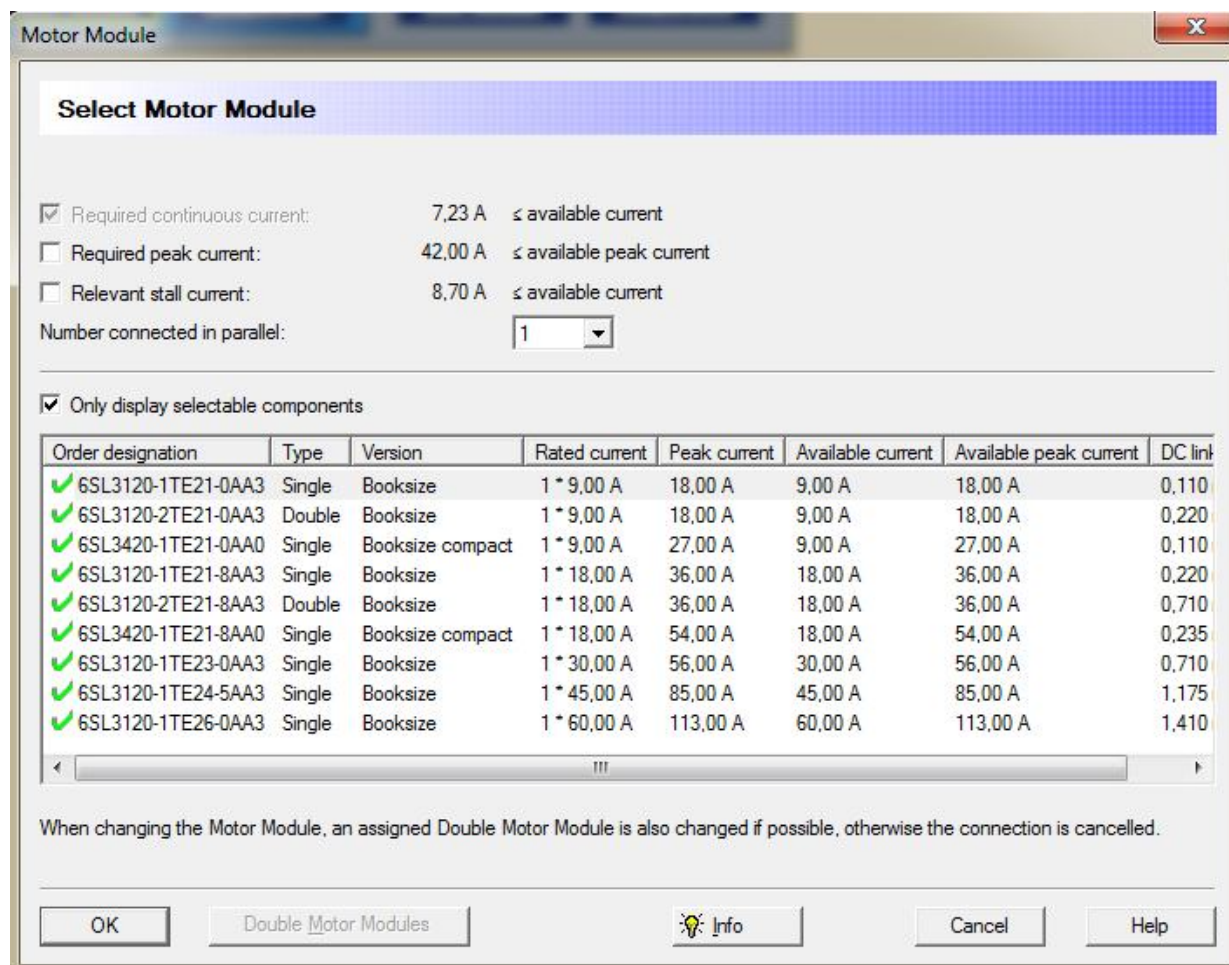


Рисунок 4.29

Виділяємо цей модуль і закриваємо вікно кнопкою ОК. На іконці Motor Module повинна з'явитися зелена галочка.

Переходимо до наступного етапу й клацаємо по іконці Output components. При цьому обновляється графічний інтерфейс і з'являється список Output options, а в полі тексту – No details available (Немає доступних подробиць). Цей список слід відкрити кнопкою ►, а потім закрити через відсутність доступних налаштувань.

При цьому після закриття вікна Output options на робочім полі стають доступними два нові списки – Supply cable (Кабель живлення) і Encoder evaluation (Визначення енкодера).

Далі кнопкою ► відкриваємо вікно Supply cable і закриваємо його без внесення змін.

Потім відкриваємо вікно Encoder evaluation, фрагмент якого показано на рисунку 4.30. У цьому вікні відображаються дані про енкодер, вбудований у двигун. Нижче у вікні розташований інтерфейс для вибору й конфігурування додаткових (зовнішніх) енкодерів, які на цей привод не ставимо.

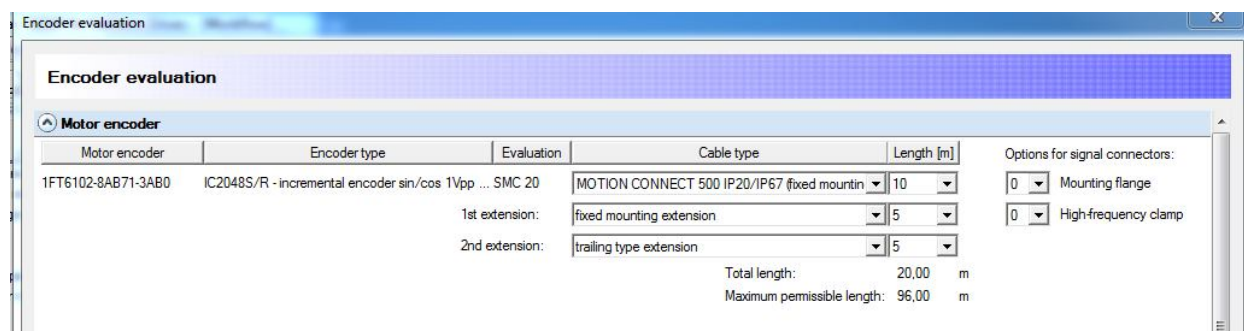


Рисунок 4.30

Закриваємо вікно Encoder evaluation, завершуючи тим самим усі налаштування привода осі X.

### **Конфігурування привода осі Z**

Далі вибираємо в графічному інтерфейсі вісь Z і клацаєм лівою кнопкою по іконці Motor.

У рядку Motor додержуємося вказівки Click here to dimension the motor і клацаємо по кнопці ►. При цьому відкривається вікно Motor wizard 1FT/1FK Step 1. Перший крок пропускаємо (двигун без редуктора) і, натиснувши кнопку Next, переходимо до наступного кроку – Basic data 1.

На цьому кроці повторюємо всі налаштування, виконані по осі X. Враховуючи те, що стояночне гальмо по осі Z завданням не передбачено, вибираємо «without holding brake». Наступні кроки виконуємо так само, як і по осі X.

Вибір модуля двигуна й вихідних компонентів для привода осі Z здійснюється аналогічно осі X.

Після завершення конфігурування осі Z переконуємося, що всі компоненти осей мають зелені галочки. Ця обставина дозволяє перейти до етапу вибору модуля живлення.

#### 4.4 Конфігурування модуля живлення

Клацаєм по іконці Line Module. При цьому інтерфейс програми здобуває вид, показаний на рисунку 4.31.

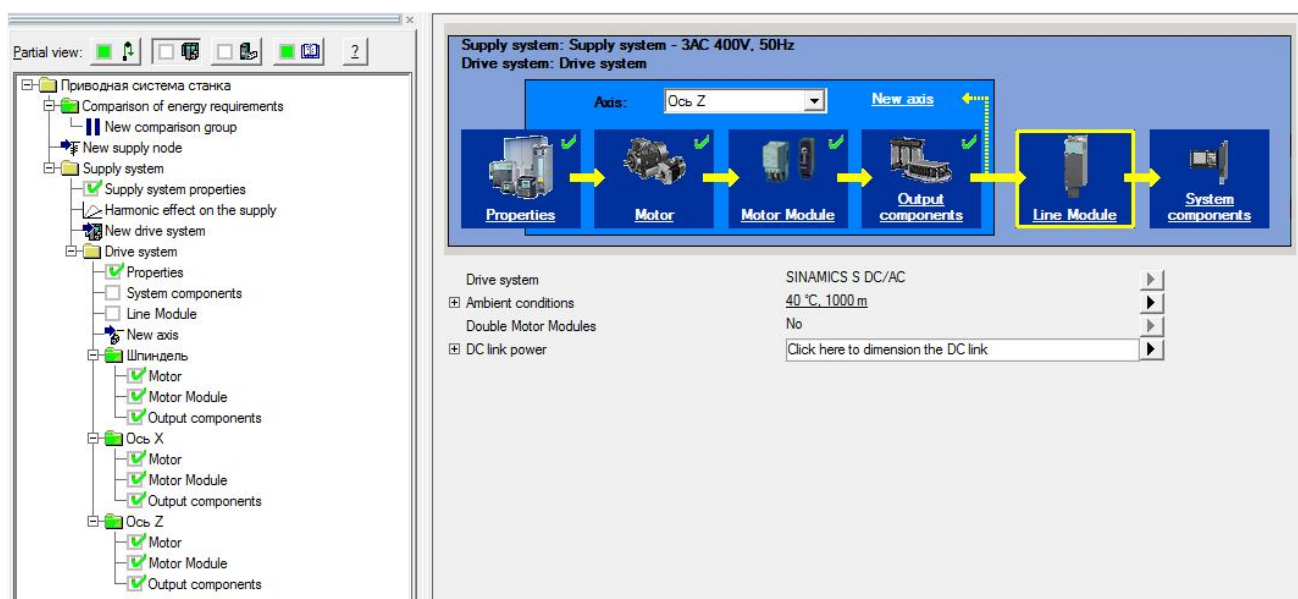


Рисунок 4.31

Далі відкриваємо кнопкою ► вікно DC link power (рис. 4.32).

У цьому вікні виконуються розрахунки необхідної потужності для лінійного модуля. Це може бути виконане шляхом розрахунків загальної потужності (коефіцієнт збігу для всієї групи осей) або за допомогою рейтингу класу потужності (коефіцієнт збігу для кожного класу потужності).

Вибір методу розрахунків проводиться перемикачами Power class rating, Total power rating і Paraxial rating.

Скористаємося методом Power class rating, суть якого полягає в тому, що приводи всіх осей розділяються на класи по потужності. Діапазони потужностей, що утворюють клас, стандартизовані. Крім того стандартизовані коефіцієнти одночасності роботи осей (Standard coincidence factor).



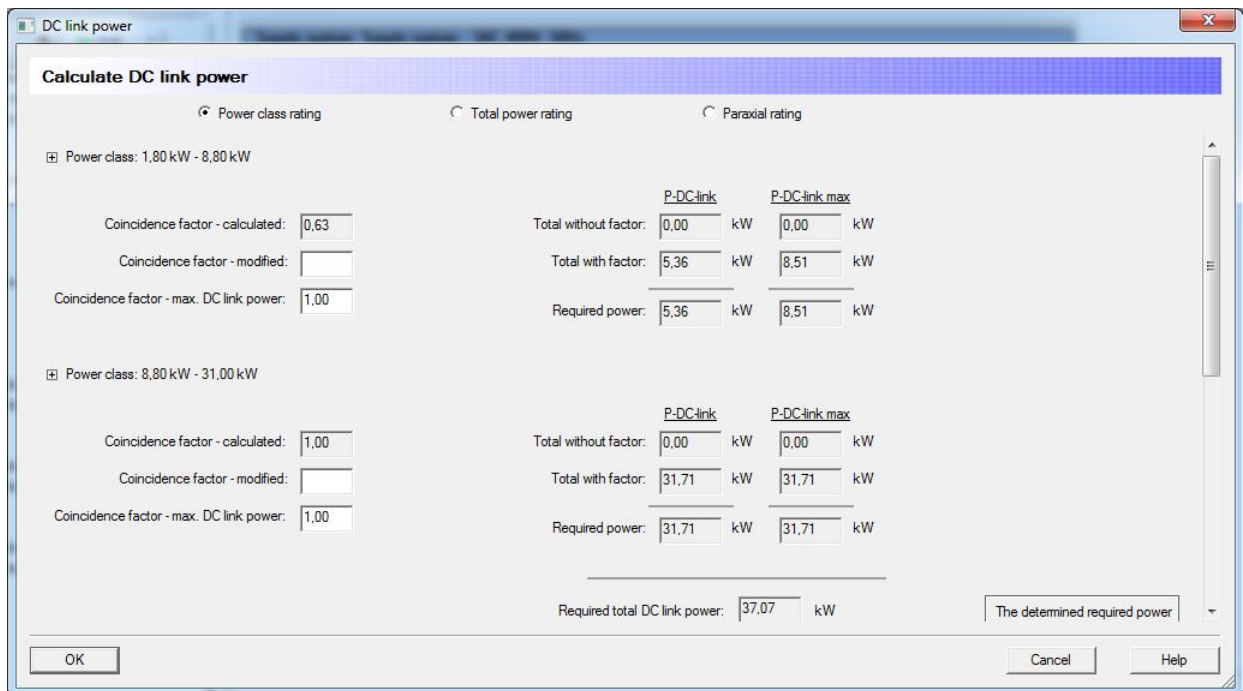


Рисунок 4.32

На рисунку 4.32 видно, що при включенні прапорця Power class rating програма SIZER виділила в проєкті два класи потужностей приводів. Перший клас визначено в діапазоні 1,8...8,8 кВт, де є два приводи подач потужністю 4,25 кВт кожний. Для цього класу стандартний, тобто розрахунковий коефіцієнт (coincidence factor) рівний 0,63. Другий клас потужності – від 8,8 до 31 кВт відповідає головному приводу. Враховуючи те, що тут один двигун, приймаємо його coincidence factor рівним 1,0.

Користувач має можливість змінити стандартний коефіцієнт у бік підвищення або у бік зменшення. Припускаючи, що робота двома приводами подач буде виконуватися не більш, ніж у половині операцій, установимо модифіковане значення коефіцієнта, рівним 0,5. Зменшимо також і значення coincidence factor до 0,8 для привода шпинделя, враховуючи, що привод головного руху зазвичай рідко навантажується до повної потужності.

Тепер клацнемо по списках Power class кожного класу потужності, щоб одержати розрахункове значення в полі Required total DC link power – 29,62 кВт, як показано на рисунку 4.33. Закриваємо це вікно кнопкою ОК. При цьому в інтерфейсі програми в рядку DC link power виводяться значення постійної й пікової потужності для вибору модуля живлення. Одночасно стає доступним новий рядок – Line Module.

Відкриваємо вікно Line Module, клацнувши по кнопці ► у цьому рядку.

У вікні Line Module, показаному на рисунку 4.34, приводяться отримані на попередньому кроці розрахункові дані по необхідній потужності джерела постійного струму (Required DC link power).

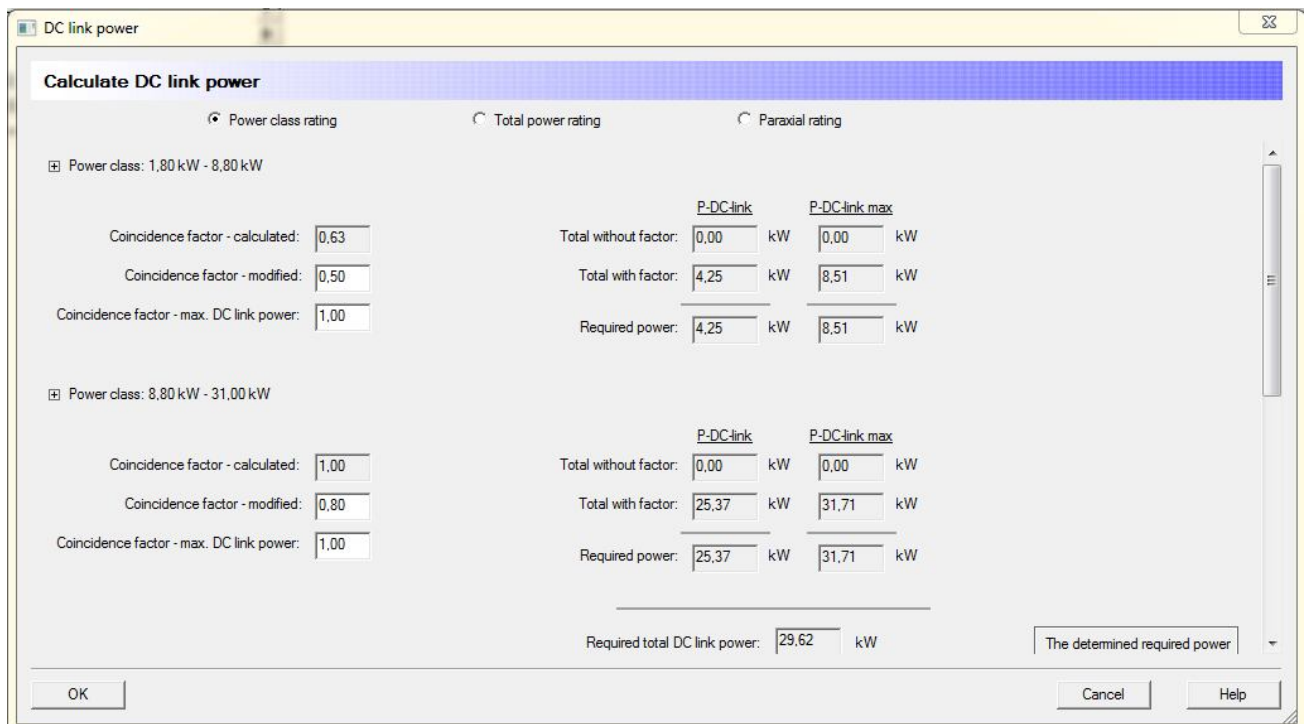


Рисунок 4.33

У нижче розташованому списку модулів живлення один з модулів підфарбований сірим кольором (перший у списку). Враховуючи те, що цей модуль повною мірою відповідає розрахунковим даним, виділяємо його й закриваємо вікно кнопкою ОК.

При цьому у вікні програми з'являється замовлений номер Line Module і стає доступним новий рядок – Braking components (Гальмові компоненти). Клацнувши по кнопці ►, відкриваємо вікно Braking components, від якого показано на рисунку 4.35.

До гальмових компонентів ставляться гальмові резистори (вони встановлюються за межами шафи) і електронні модулі, які підключають ці резистори до проміжного контуру. Гальмові компоненти потрібні для керованої зупинки привода при відключенні живильної мережі (аварійний останов), а також при динамічній гальмуванні, при яким відбувається рекуперація енергії навантаження і її розсіювання у вигляді тепла на гальмовому резисторі. У вікні Braking components виведений список гальмових компонентів для установки в приводну систему.

Слід урахувати, що для вибору конкретного резистора необхідно знати параметри швидкості й часу в циклах гальмування. Приймаючи до уваги, що ці параметри невідомі, вікно Braking components закриваємо, не виконуючи ніяких дій. На іконці Line Module з'являється зелена галочка, що свідчить про закінчення налаштування цього модуля.

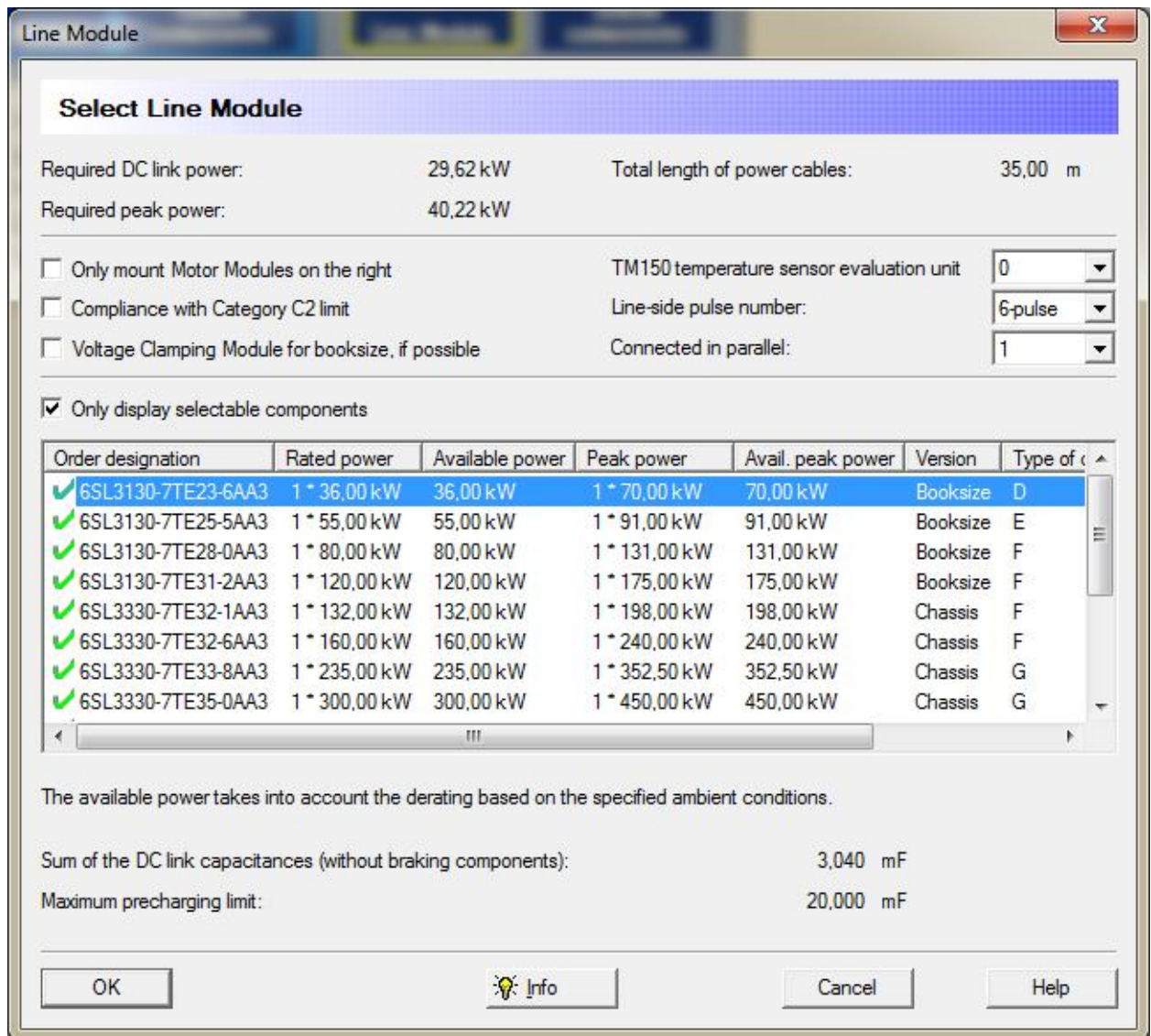


Рисунок 4.34

#### 4.4 Вибір системних компонентів

Переходимо до наступного етапу – клацаємо по іконці System components. При цьому обновляється графічний інтерфейс і з'являється список Input options. Клацнувши по кнопці ►, відкриваємо вікно Input options, у якому, як опція включене застосування активного модуля живлення. Закриваємо вікно, при цьому стає доступним новий список – Additional accessories (Додаткові аксесуари).

Клацнувши по кнопці ►, відкриваємо вікно Accessories, у якому нічого не вибираємо через непотрібність. Закриваємо це вікно, у результаті з'являються два нові рядки – Open-loop/closed-loop control ... і Additional components.

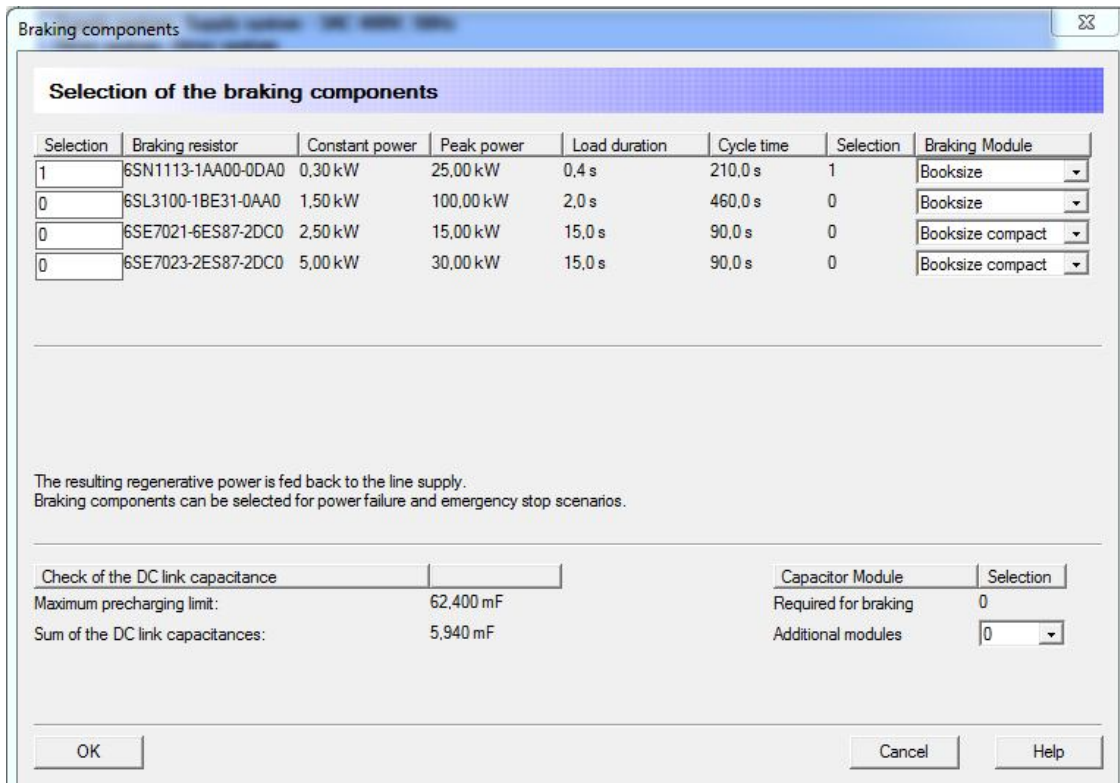


Рисунок 4.35

Кнопкою ► у рядку Open-loop/closed-loop control ... перемикаємо вид Partial view в **Open-loop/closed-loop control / 24 V / Cabinet Module**, як показано на рисунку 4.36.

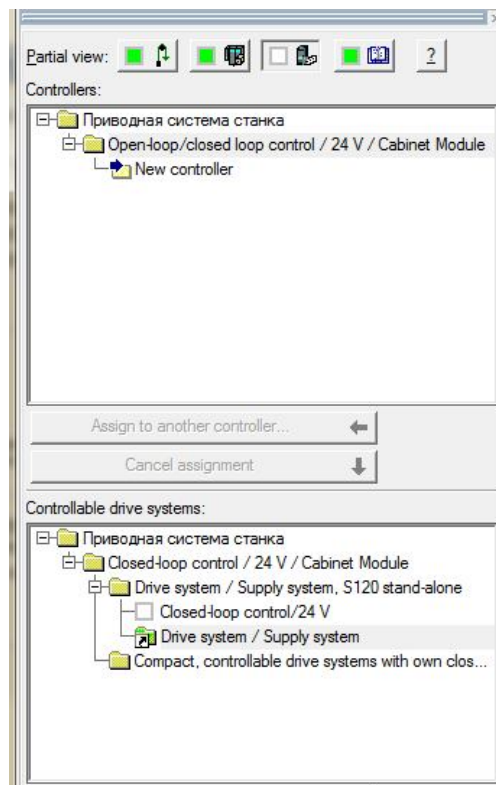


Рисунок 4.36



Подвійним клацанням по завданню New controller у дереві проекту відкриваємо вікно Add controller, яке показано на рисунку 4.37. Із трьох систем керування вибираємо систему ЧПУ SINUMERIK і закриваємо вікно кнопкою ОК. У робочій області з'являється нова вистава інтерфейсу, показана на рисунку 4.38.

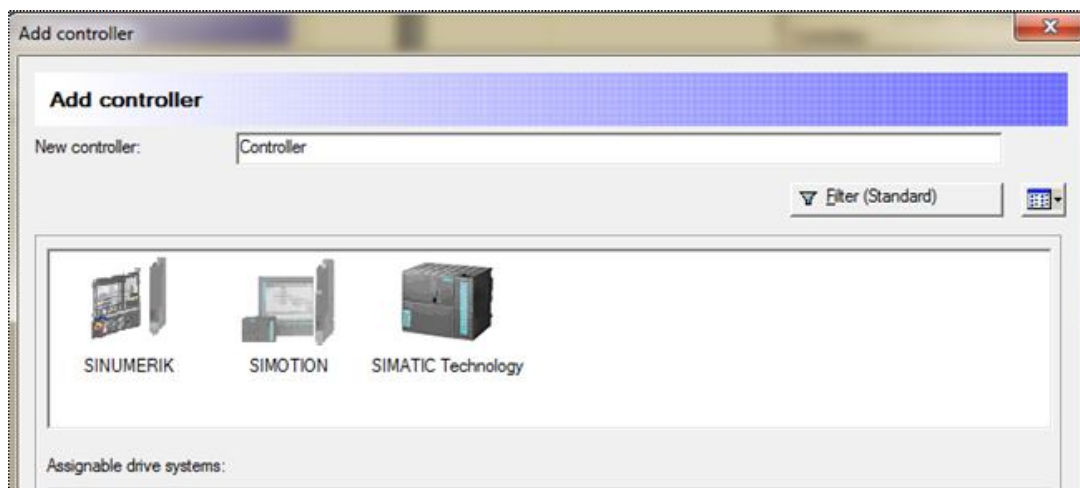


Рисунок 4.37



Рисунок 4.38

Натискаємо кнопку ► у рядку Open-loop/closed-loop control electronics і відкриваємо вікно з тим же іменем (рис. 4.39).

У вікні Open-loop/closed-loop control electronics вибираємо тип системи SINUMERIK – 840DsI, далі процесор ЧПУ типу NCU 710.2, потім термінальний процесор PCU 50.3B-C для панелі оператора, а також програмне забезпечення панелі оператора – SINUMERIK Operate.

Не закриваючи вікна, перемикаємося з виду Topology на вид SINUMERIK, як показано на рисунку 4.40.

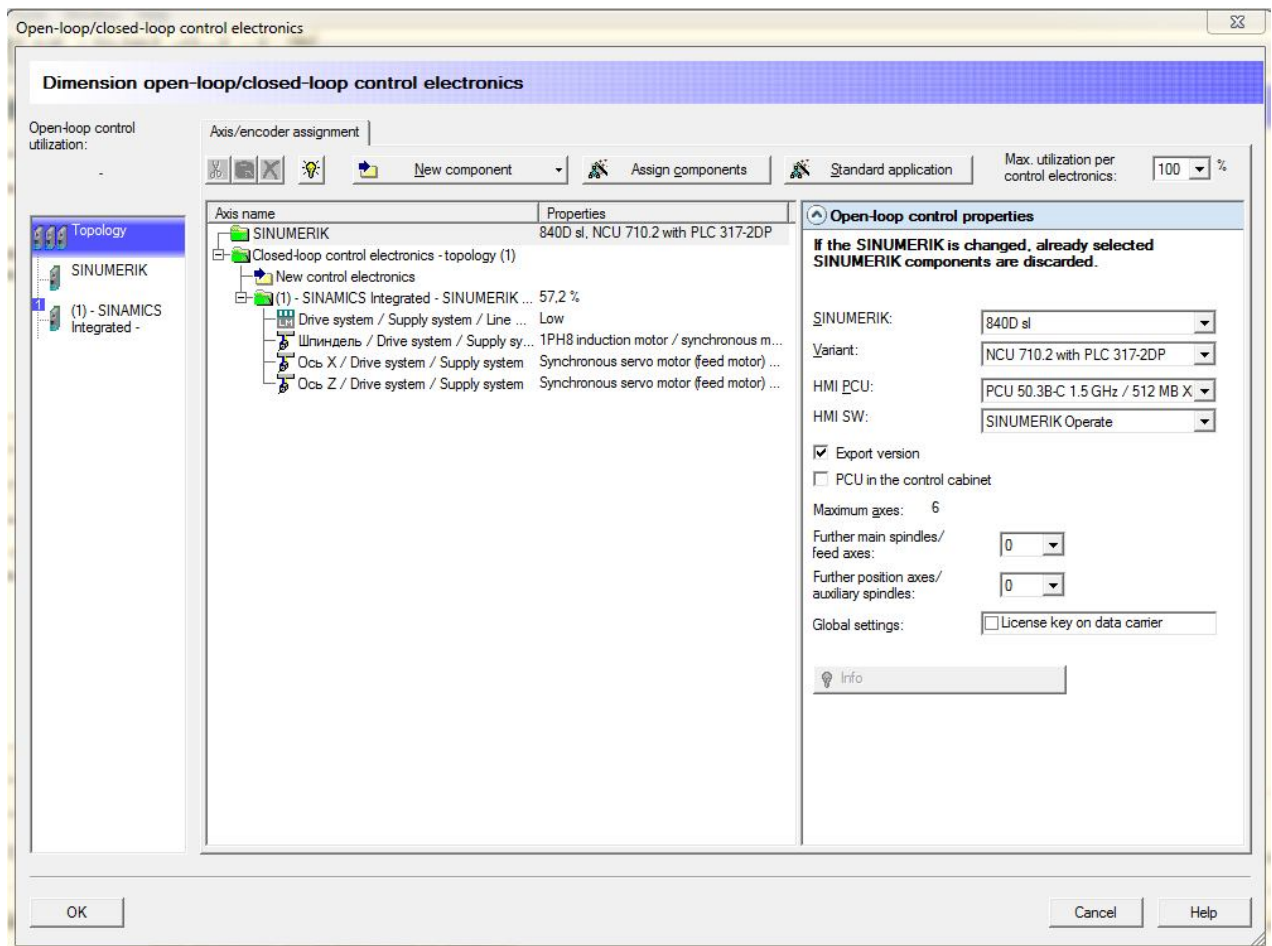


Рисунок 4.39

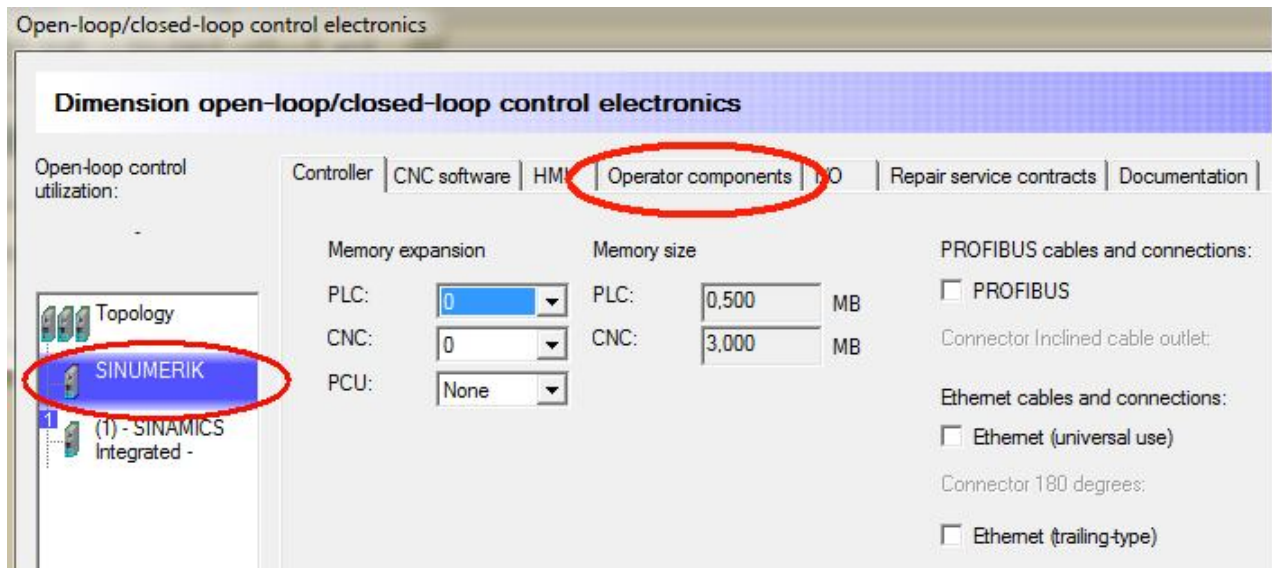
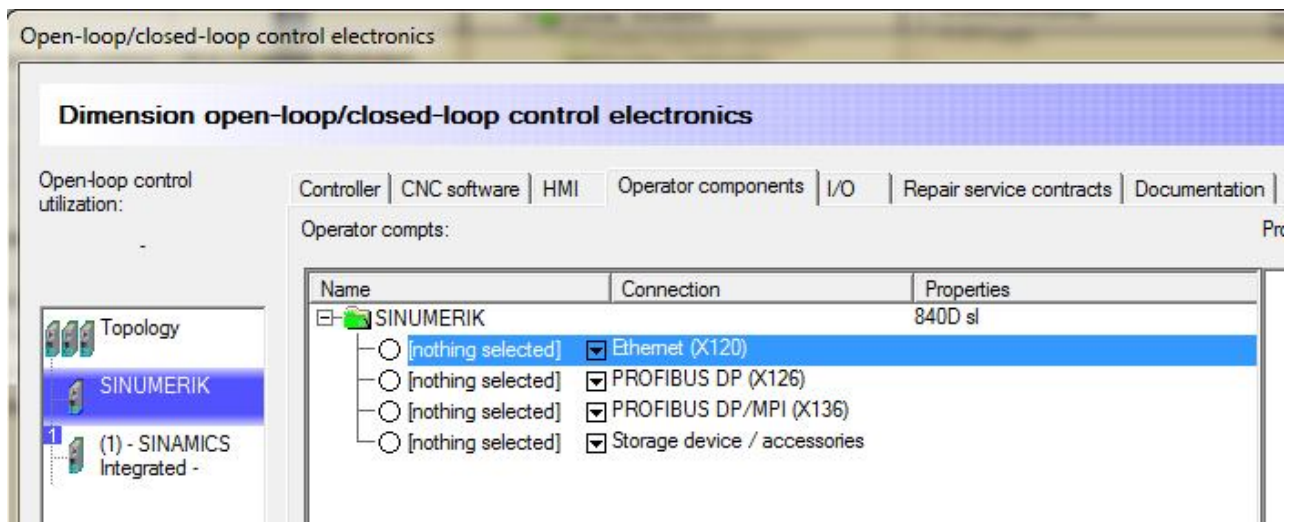


Рисунок 4.40

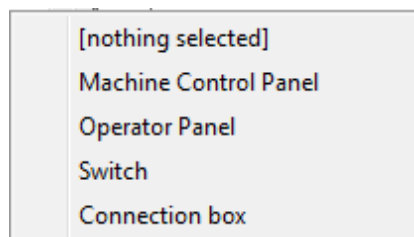
Тут потрібно перейти із вкладки Controller на вкладку Operator components, від якої показано на рисунку 4.41.



*Рисунок 4.41*

На вкладці Operator components у стовпці Connection вибираємо послідовно рядки Ethernet і PROFIBUS, у яких кнопкою ▼ розкриваємо список можливих підключень компонентів керування.

Так, наприклад, на рисунку 4.42 показаний список, відкритий у рядку Ethernet (X120).



*Рисунок 4.42*

У результаті таких послідовних дій проводиться вибір панелі оператора (Operator Panel OP 012), верстатного пульта (Machine Control Panel MCP 310), а також клавіатури ЧПУ (Keyboard KB 310C). Обрані компоненти й засоби інтерфейсу для їхнього підключення показано на рисунку 4.43.

Закінчивши вибір компонентів керування, закриваємо вікно Open-loop/closed-loop control electronics кнопкою ОК. У результаті в меню завдань **Open-loop/closed-loop control / 24 V / Cabinet Module** розділу проекту Controller, SINUMERIK стає доступним нове завдання (рис. 4.44) – Installation arrangement (Розташування установки).

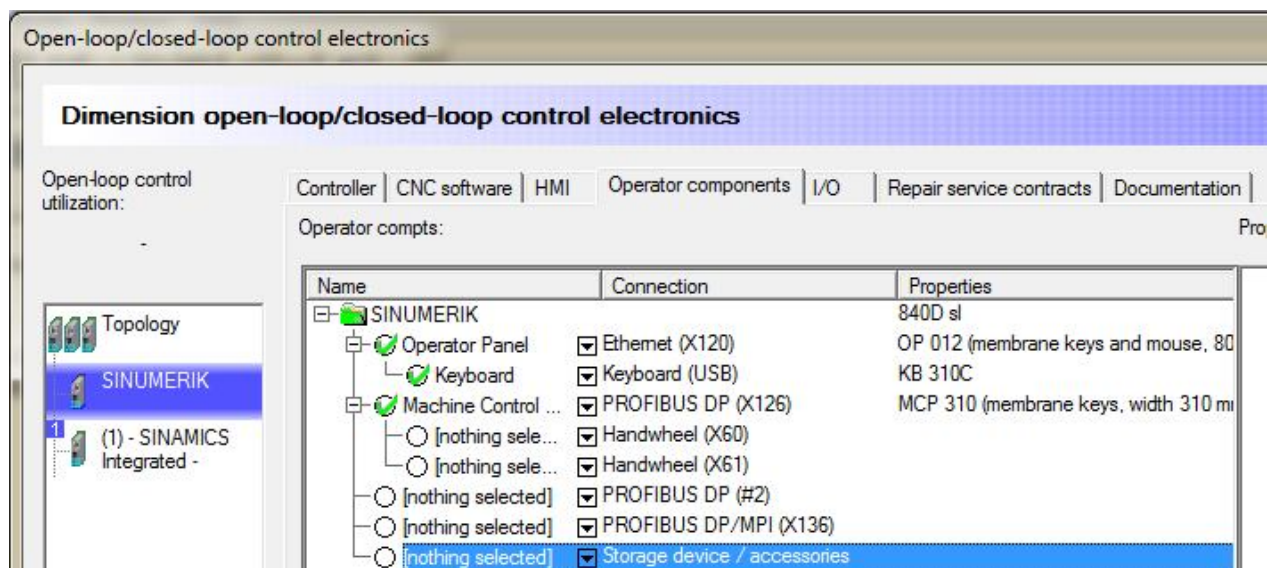


Рисунок 4.43

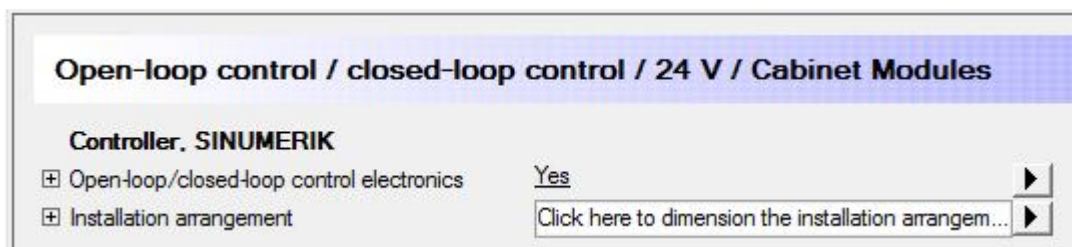


Рисунок 4.44

Тут тиснемо на кнопку ► у рядку Installation arrangement, у результаті відкривається вікно з тим же іменем, показане на рисунку 4.45.

У цьому вікні представлені дані по інтегрованій системі SINUMERIK-SINAMICS. Переконавшись у повноті системи, тобто в тому, що присутні всі модулі системи, закриваємо вікно кнопкою ОК.

Після закриття вікна Installation arrangement стає доступним нове завдання – Drive-cliq topology (рис. 4.46).

Тиснемо на кнопку ► у рядку Drive-cliq topology. При цьому відкривається однойменне вікно (рис. 4.47), у якому представлена топологічна схема з'єднання ЧПУ-процесора NCU 710.2 з модулями приводної системи.

Якщо всі необхідні з'єднання існують, то кнопкою Next переходимо до наступної виду, показаному на рисунку 4.48. Тут наведена більш докладна інформація із з'єднань модулів системи. Спочатку звертаємо увагу на колір ліній з'єднань. Так, наприклад, на схемі з'єднань рисунка 4.48 частина кабелів пофарбована в чорний колір, а частина в сірий. Чорний колір кабелю означає, що він не визначений.



Дійсно, якщо подивитися на список кабелів у нижній частині вікна, то можна побачити, що всі кабелі з рознімачів X202 моторних модулів (SMM) не визначені.

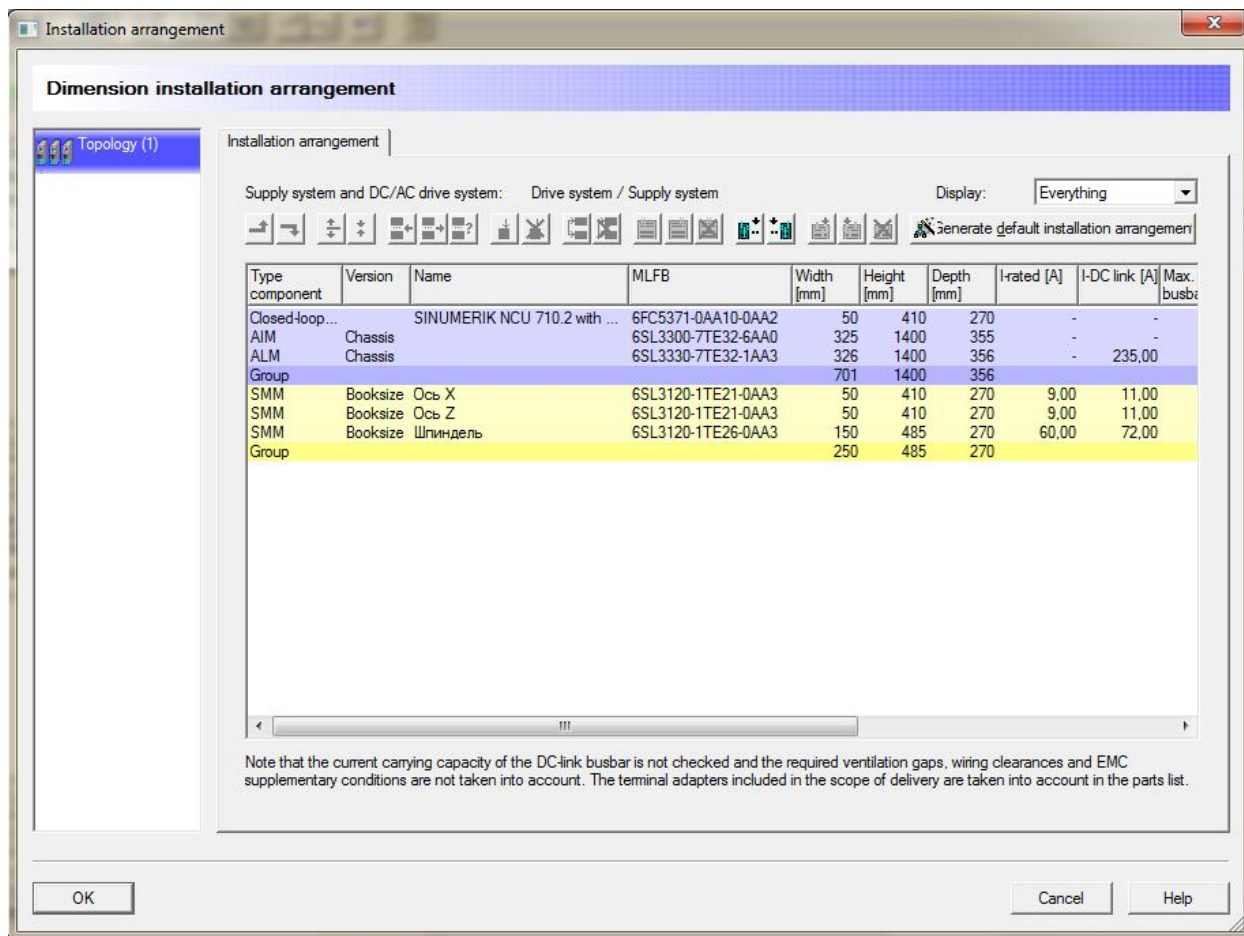


Рисунок 4.45

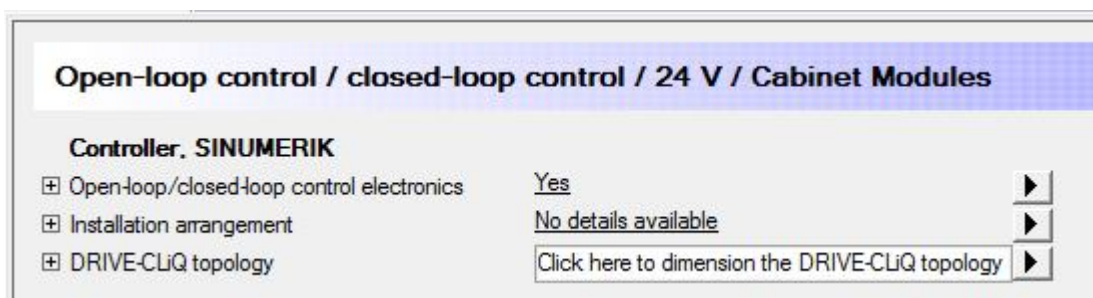


Рисунок 4.46

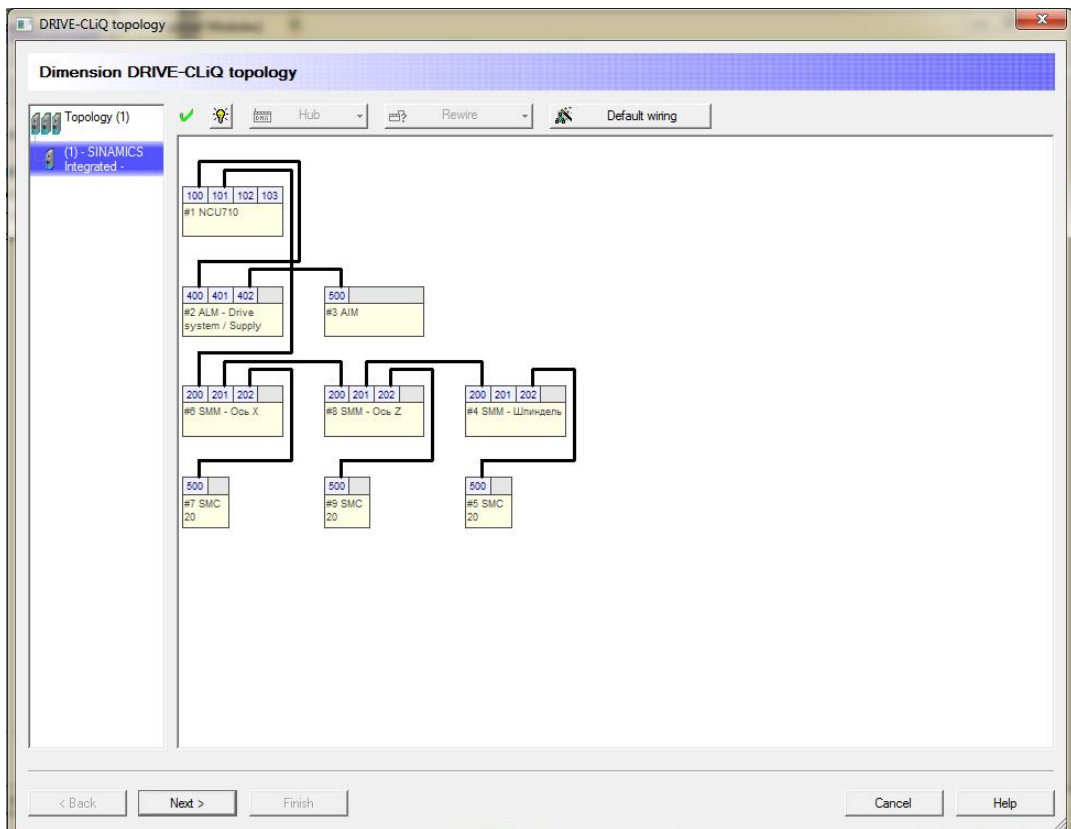


Рисунок 4.47

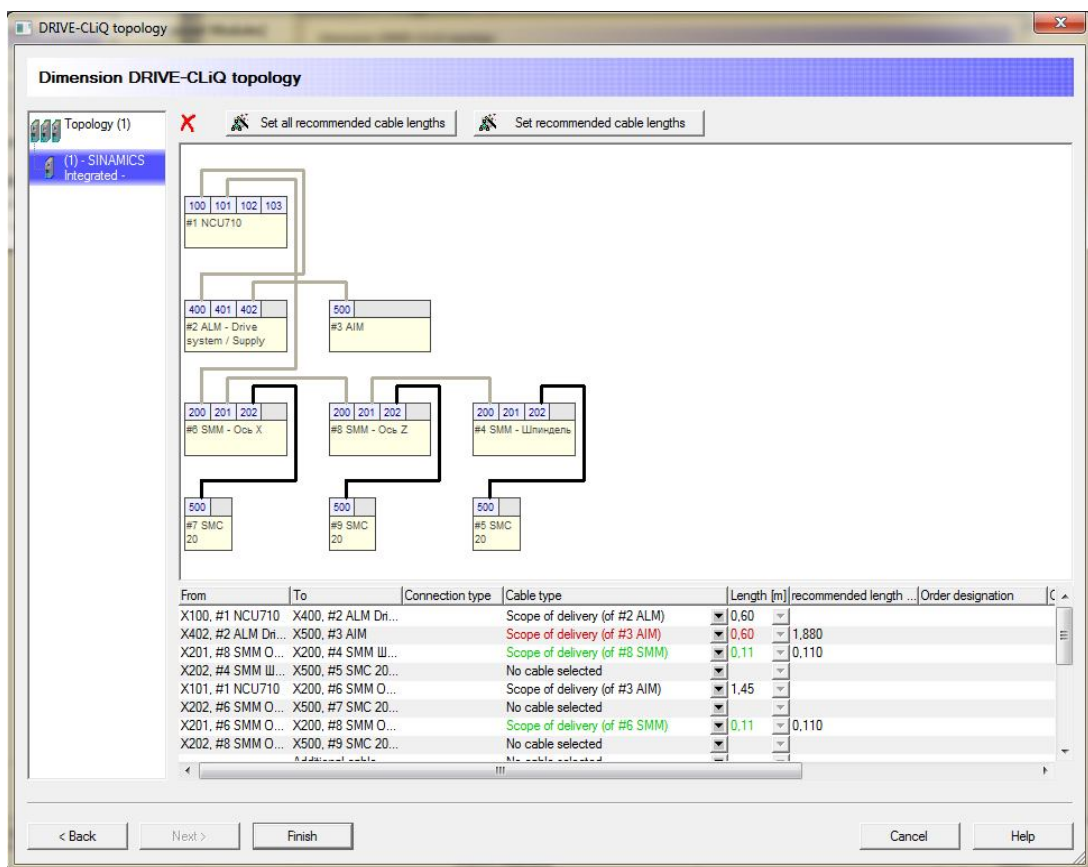


Рисунок 4.48

Для визначення кабелю необхідно клацнути по кнопці ▼ у колонку Cable type і в списку, що розкрився, показаному на рисунку 4.49, вибрати тип кабелю, наприклад, DRIVE-CLiQ cable (by the meter) IP20/IP20 (fixed mounting).

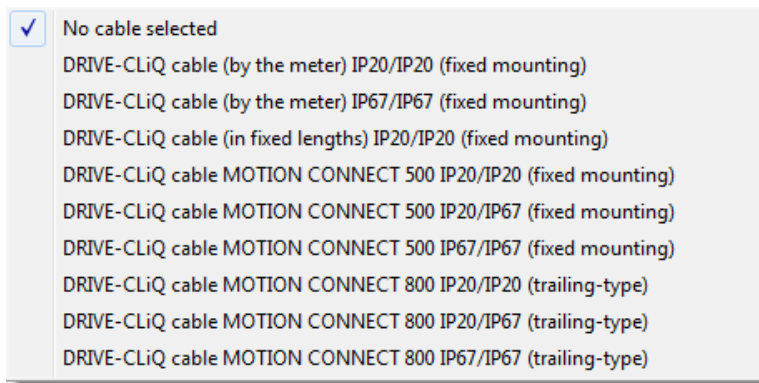


Рисунок 4.49

Після вибору кабелю в стовпці Length біля кнопки ▼ з'являється червоний трикутник. Відкриваємо цією кнопкою список довжин кабелів і призначаємо яку-небудь довжину. На цьому визначення кабелю завершується.

Після визначення всіх кабелів вікно Drive-cliq topology здобуває вид, показаний на рисунку 4.50.

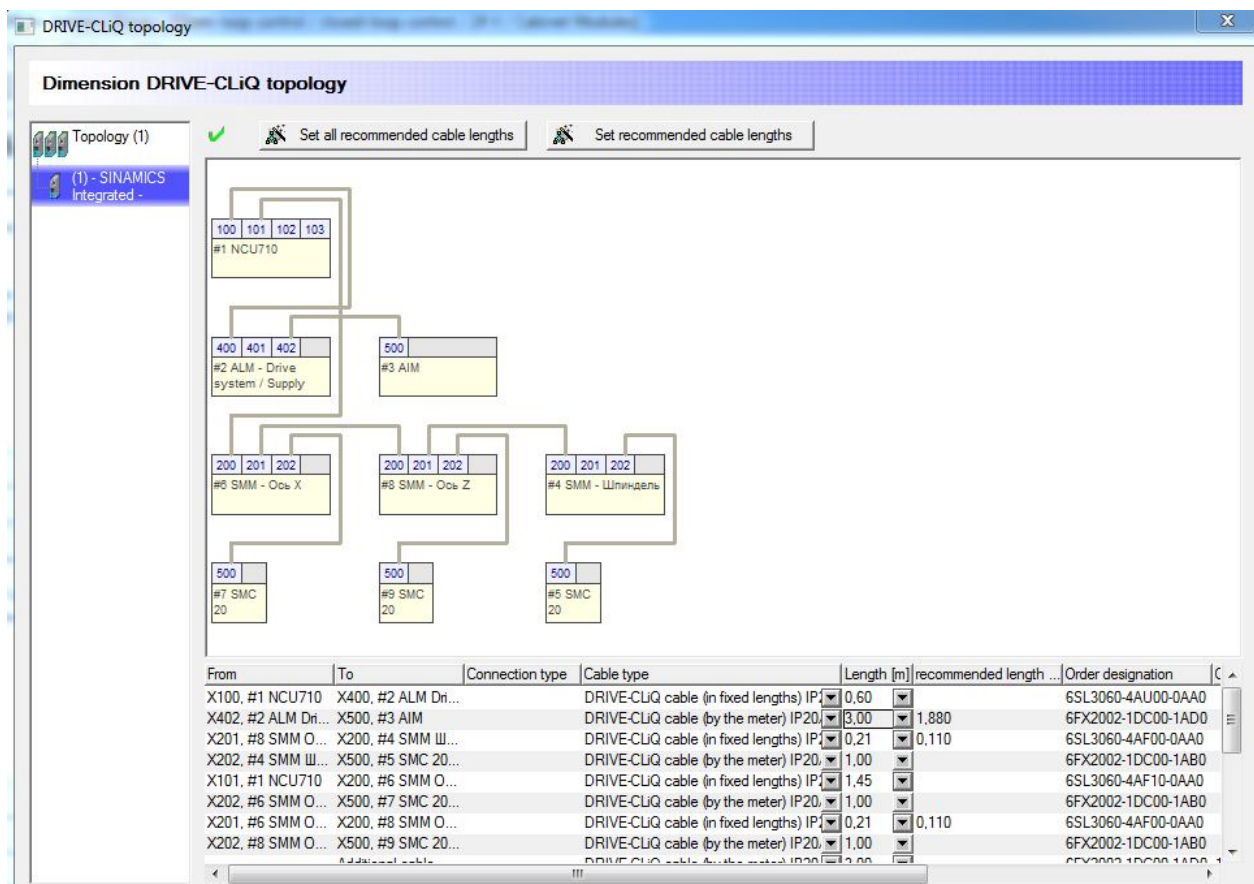


Рисунок 4.50

Закриваємо вікно кнопкою ОК. При цьому стає доступним нове завдання – 24 V supply (рис. 4.51).



Рисунок 4.51

Натискаємо кнопку ► у рядку 24 V supply і переходимо у вікно 24 V power supply, від якого показано на рисунку 4.52.

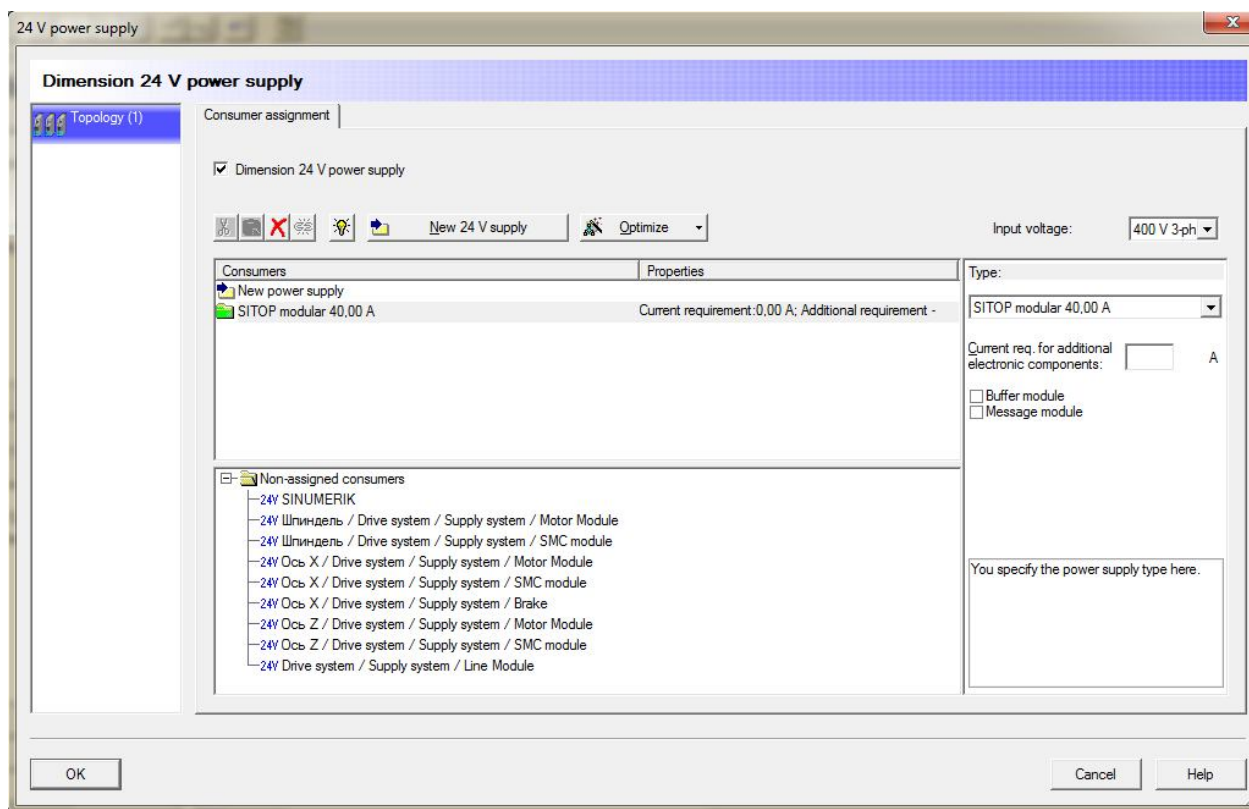


Рисунок 4.52

У нижній частині вікна 24 V power supply показаний список споживачів (якщо список частково схований, то його можна підняти за допомогою лівої кнопки миші, зачепивши розділову лінію). У верхній частині вікна зазначене рекомендоване джерело живлення.

Якщо споживачі не призначені цьому блоку живлення, то в заголовку списку споживачів зазначене: Non-assigned consumers (Не призначені



споживачі). Для призначення споживачів потрібно відкрити кнопкою Optimize список блоків живлення. У списку показані блоки на 5 А, 10 А, 20 А и на 40 А два типи – modular (модульна конструкція) і smart (компактна конструкція). Вибираємо більш досконалий модуль SITOP modular з рекомендованим значенням сили струму 40,00 А. При цьому виводиться попередження: Do you really want to optimize? The existing assignment will be lost (Ви дійсно прагнете оптимізувати? Поточні призначення будуть загублені).

Підтверджуємо намір кнопкою ОК, у результаті всі споживачі підключаються до модуля живлення із вказівкою споживаного струму, як показано на рисунку 4.53.

Існує також можливість призначення менш потужних модулів живлення, наприклад, два модулі по 20 А, або інші комбінації.

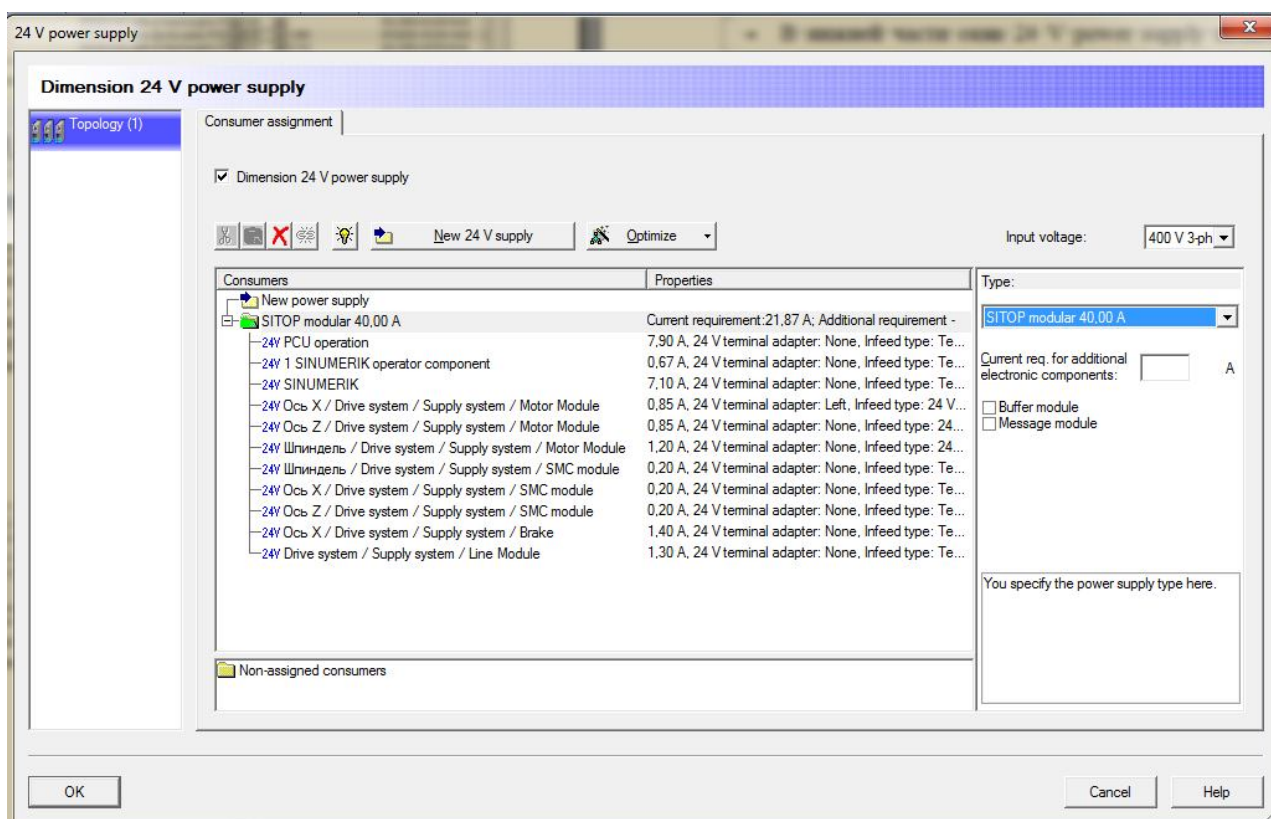


Рисунок 4.53

Закриваємо вікно 24 V power supply кнопкою ОК, завершуючи тим самим процес вибору системних компонентів і проектування приводної системи.

На закінчення необхідно постачити проект відомостями про автора. Для цього в меню Project вибираємо Project information і у вікні, що відкрилося, Project info, показаному на рисунку 4.54, уводимо найменування кафедри (Department of APP), прізвище й групу автора, а в полі коментаря найменування дисципліни: *Цифрові системи керування й обробки інформації*.

*Рисунок 4.54*

Для перегляду результатів проектування перемикаємося з виду Project на вид Results, як показано на рисунку 4.55.

Тут проект представлений як структура технічної документації. Двічі клацнувши по обраному рядку, можна ознайомитися з технічними даними по будь-якому компоненту системи.

#### **4.5 Індивідуальні завдання й вимоги до звіту по роботі**

Індивідуальні завдання наведені в додатку В. Номер варіанта визначається порядковим номером прізвища студента в журналі академічної групи.

Звіт по роботі повинен містити:

- завдання на конфігурування приводної системи верстата із ЧПУ;
- роздруковка наступних розділів з технічної документації:
  1. Technical data
  2. Structure for SINAMICS S120 drive system;
  3. 24 V supple;



4. Operator components;
5. Drive-cliq topologies;

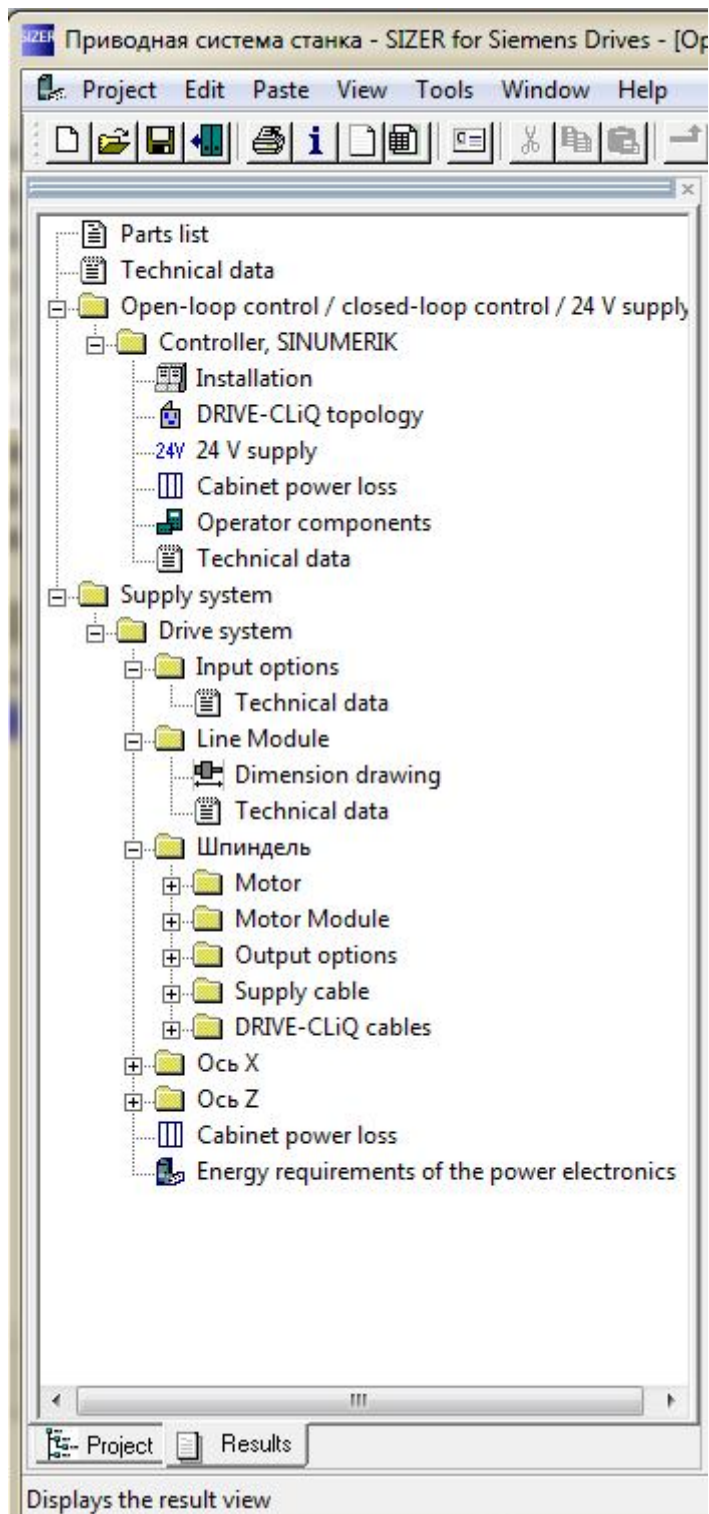


Рисунок 4.55

Роздруківка документації проводиться в такий спосіб. Відкриваємо меню View і вибираємо опцію Display result view. Далі натискаємо на кнопку Print (Принтер) у меню інструментів. При цьому відкривається вікно Print selection,

показане на рисунку 4.56. Якщо на комп'ютері встановлена програма віртуального принтера doPDF v7, то в полі Printer вона з'являється за замовчуванням. Для вибору іншого принтера тиснемо кнопку Select.

Далі у вікні Print selection ставимо галочки в необхідних розділах (Chapter) документації й натискаємо ОК (рис. 4.56).

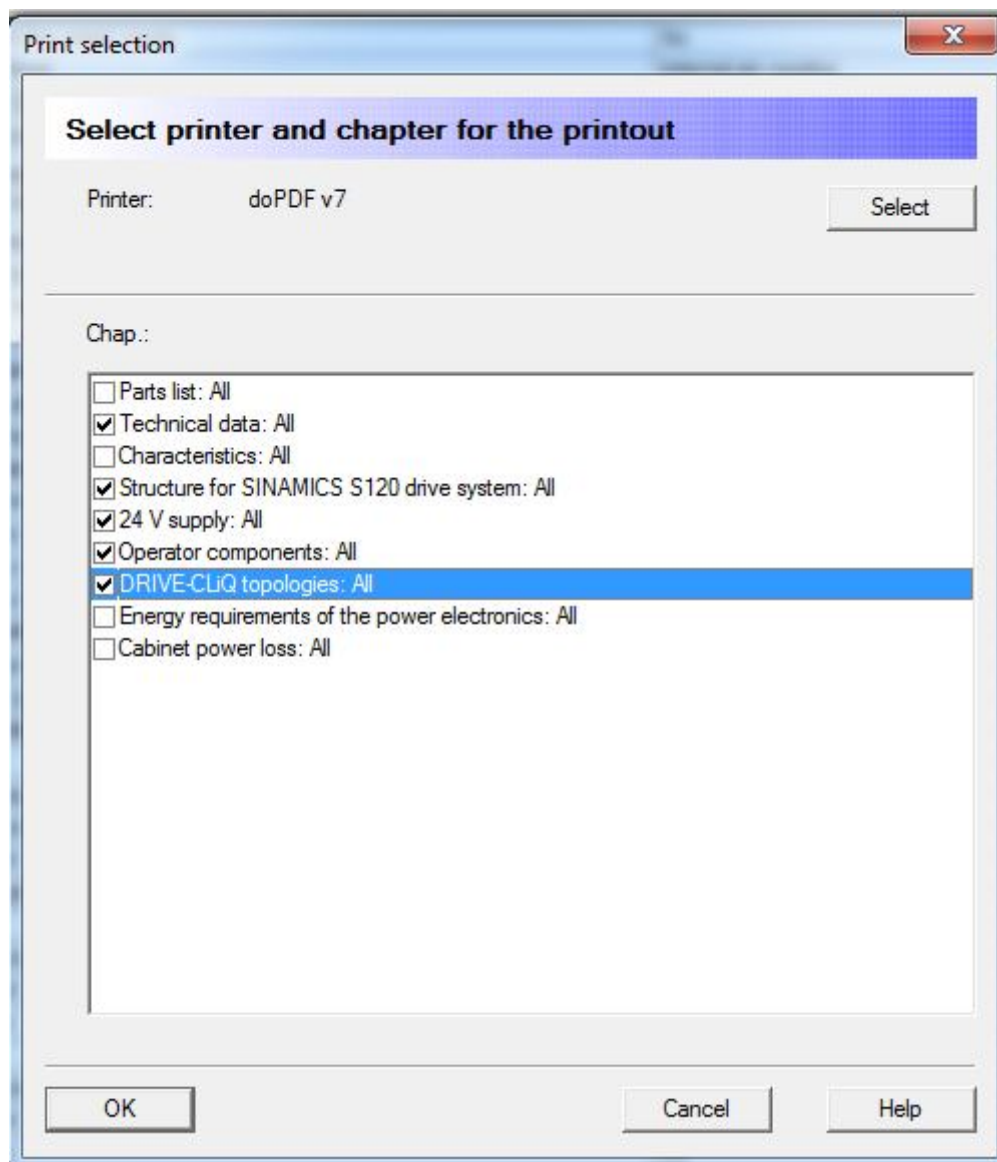


Рисунок 4.56

При використанні програми doPDF v7 відкривається вікно збереження файлу (формат PDF), показане на рисунку 4.57, у якому треба вказати місце збереження файлу і його ім'я.

Після закриття цього вікна проводиться конвертування файлу у формат PDF і, якщо встановлена галочка в полі «Відкрити PDF», через кілька секунд файл з'являється на екрані. Фрагмент першої сторінки файлу показано на рисунку 4.58.

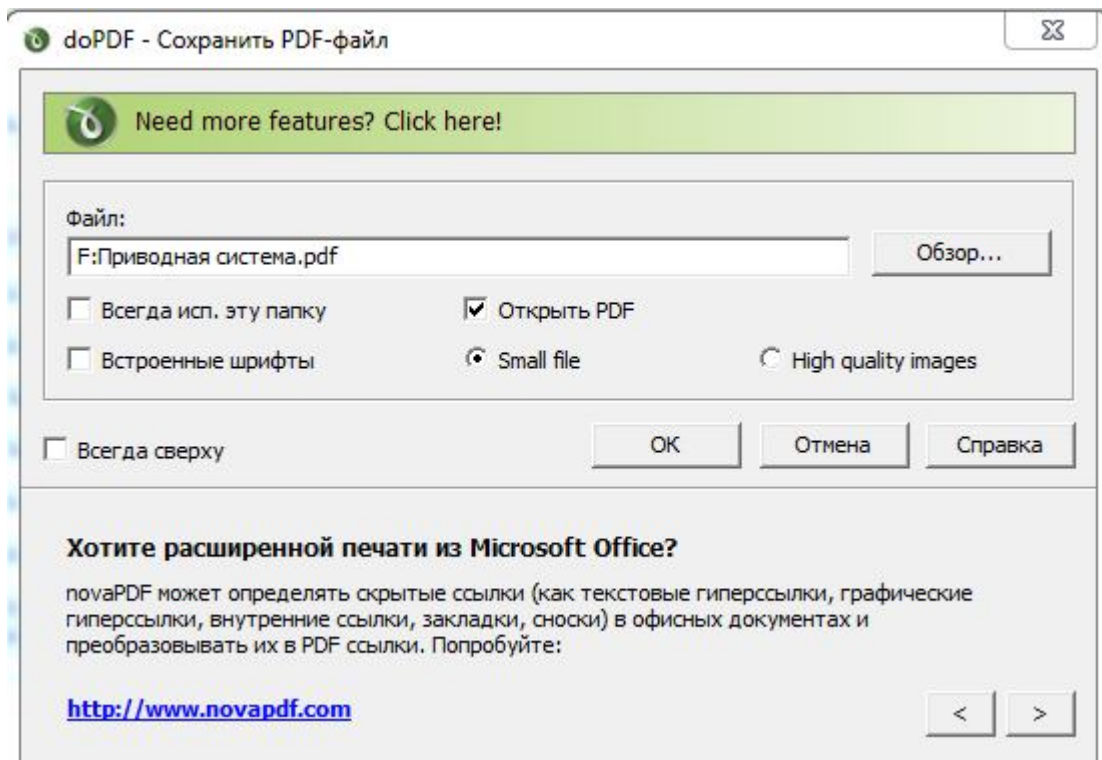


Рисунок 4.57

# SIEMENS

17.12.2014

Customer:  
 Plant: Department of APP

Customer No.:  
 Last saved:  
 Project: Приводная система станка  
 Author: Иванов И. И. гр.АПП 11-1  
 Comment: Цифровые системы управления и обработки информации

## 1. Technical data

### • Supply system

#### - Line data

Voltage	400 V
Frequency	50 Hz
Number of phases	3
Allowance for differing supply voltage	No
Allowance for short term supply fluctuations	Yes
Maximum temporary undervoltage to the rated voltage	15 %
Undervoltage	340 V

Рисунок 4.58

Для печати всієї документації по проекту розкриваємо меню Project і вибираємо Print. Обсяг печати становить 18 аркушів.

## ДОДАТОК А

### Базові функції STL

#### Двійкові логічні операції

A	-	операція І (AND) для перевірки присутності рівня "1"
AN	-	операція І (AND) для перевірки присутності рівня "0"
O	-	операція АБО (OR) для перевірки присутності рівня "1"
ON	-	операція АБО (OR) для перевірки присутності рівня "0"
X		операція АБО виключаюче (Exclusive OR) для перевірки - рівня "1"
XN		операція АБО виключаюче (Exclusive OR) для перевірки рівня "0"
-	I	вхід
-	Q	вихід
-	M	меркер
-	L	біт в області локальних даних
-	T	функція таймера
-	C	функція лічильника
-	DBX	біт в області глобальних даних
-	DIX	біт в екземплярному блоку даних DB
-	==0	значення результату операції дорівнює нулю
-	<>0	значення результату операції не дорівнює нулю
-	>0	значення результату операції більше нуля
-	>=0	значення результату операції більше нуля або дорівнює нулю
-	<0	значення результату операції менше нуля
-	<=0	значення результату операції менше нуля або дорівнює нулю
-	UO	значення результату операції невірно
-	OV	переповнення
-	OS	збережене переповнення
-	BR	двійковий результат
A(		операція І (AND) з відкриваючою дужкою
AN(		операція І (AND) з відкриваючою дужкою
O(		операція АБО (OR) з відкриваючою дужкою
ON(		операція АБО (OR) з відкриваючою дужкою
X(		операція АБО виключаюче (Exclusive OR) з відкриваючою дужкою
XN(		операція АБО виключаюче (Exclusive OR) з відкриваючою дужкою
)		закриваюча дужка
O		операція АБО (OR), що поєднує операції І (AND)
NOT		операція заперечення RLO
SET		операція установки RLO
CLR		операція скидання RLO
SAVE		операція фіксації RLO в BR

#### Операції з пам'яттю

=	-	операція присвоєння
S	-	операція установки
R	-	операція скидання
FP	-	позитивний фронт сигналу
FN	-	негативний фронт сигналу
-	I	вхід
-	Q	вихід
-	M	меркер
-	L	біт в області локальних даних
-	DBX	біт в області глобальних даних

- DIX біт в екземплярному DB

### **Функції передачі**

L - операція завантаження (load)

T - операція передачі (transfer)

- IB вхідний байт

- IW вхідне слово

- ID вхідне подвійне слово

- QB вихідний байт

- QW вихідне слово

- QD вихідне подвійне слово

- MB байт меркерів

- MW слово меркерів

- MD подвійне слово меркерів

- LB байт локальних даних

- LW слово локальних даних

- LD подвійне слово локальних даних

- DBB байт глобальних даних

- DBW слово глобальних даних

- DBD подвійне слово глобальних даних

- DIB байт в екземплярному DB

- DIW слово в екземплярному DB

- DID подвійне слово в екземплярному DB

- STW слово стану

LPIB завантаження (load) периферійного вхідного байта

LPIW завантаження (load) периферійного вхідного слова

LPID завантаження (load) периферійного вхідного подвійного слова

TPQB передача (transfer) периферійного вихідного байта

TPQW передача (transfer) периферійного вихідного слова

TPQD передача (transfer) периферійного вихідного подвійного слова

L T "звичайна" завантаження значення таймера

LCT завантаження значення таймера в BCD-кодi

LC "звичайна" завантаження значення лічильника

LCC завантаження значення лічильника в BCD-кодi

L *const* завантаження (load) константи

L P#.. завантаження (load) покажчика

L P#var завантаження (load) початкової адреси змінної

### **Функції акумуляторів**

PUSH зрушення вмісту акумуляторів "уперед"

POP зрушення вмісту акумуляторів "назад"

ENT зрушення вмісту акумуляторів 2 і 3 "уперед"

LEAVE зрушення вмісту акумуляторів 3 і 4 "уперед"

TAK обмін умістом між акумуляторами 1 і 2

SAW обмін умістом між байтами 0 і 1 акумулятора 1

CAD обмін умістом між усіма байтами акумулятора 1

### **Функції таймерів**

SPT запуск таймера в режимі "керованого імпульсу"

SET запуск таймера в режимі "розширеного імпульсу"

SDT запуск таймера в режимі "із затримкою включення"

SST запуск таймера в режимі "із затримкою включення з пам'яттю"

SFT запуск таймера в режимі "із затримкою вимикання"

R T скидання таймера

FR T дозвіл перезавантаження таймера

### **Функції лічильників**

CU C запуск лічильника в режимі "прямий рахунок"

CD C запуск лічильника в режимі "зворотний рахунок"

S C установка лічильника

R C скидання лічильника

FR C дозвіл перезавантаження лічильника

### **Функції для обробки чисел**

#### **Функції порівняння**

==I перевірка даних формату INT на рівність

<>I перевірка даних формату INT на нерівність

>I порівняння даних формату INT за критерієм "більше чим"

>=I порівняння даних формату INT за критерієм "більше або рівно"

<I порівняння даних формату INT за критерієм "менше ніж"

<=I порівняння даних формату INT за критерієм "менше або рівно"

==D перевірка даних формату DINT на рівність

<>D перевірка даних формату DINT на нерівність

>D порівняння даних формату DINT за критерієм "більше чим"

>=D порівняння даних формату DINT за критерієм "більше або рівно"

<D порівняння даних формату DINT за критерієм "менше ніж"

<=D порівняння даних формату DINT за критерієм "менше або рівно"

==R перевірка даних формату REAL на рівність

<>R перевірка даних формату REAL на нерівність

>R порівняння даних формату REAL за критерієм "більше чим"

>=R порівняння даних формату REAL за критерієм "більше або рівно"

<R порівняння даних формату REAL за критерієм "менше ніж"

<=R порівняння даних формату REAL за критерієм "менше або рівно"

#### **Математичні функції**

SIN синус

COS косинус

TAN тангенс

ASIN арксинус

ACOS арккосинус

ATAN арктангенс

SQR знаходження квадрата числа

SQRT добування квадратного кореня із числа

EXP експонента по підставі  $e$

LN натуральний логарифм

#### **Арифметичні функції**

+I додавання двох чисел формату INT

-I вирахування двох чисел формату INT

\*I множення двох чисел формату INT

/I розподіл двох чисел формату INT

+D додавання двох чисел формату DINT

-D вирахування двох чисел формату DINT

\*D множення двох чисел формату DINT

/D розподіл двох чисел формату DINT (ціла частина)

MOD розподіл двох чисел формату DINT (залишок)

+R додавання двох чисел формату REAL

-R вирахування двох чисел формату REAL

\*R множення двох чисел формату REAL



/R розподіл двох чисел формату REAL  
 + *const* додавання з константою  
 + R#.. додавання з покажчиком  
 DEC *n* декрементування  
 INC *n* інкрементування

#### Функції перетворення

ITD конвертування даних формату INT у формат DINT  
 ITB конвертування даних формату INT у формат BCD  
 DTB конвертування даних формату DINT у формат DINT  
 DTR конвертування даних формату DINT у формат REAL  
 BTI конвертування даних формату BCD у формат INT  
 BTD конвертування даних формату BCD у формат DINT  
 Конвертування даних формату REAL у формат DINT, при цьому відбувається:  
 RND+ округлення даних до найближчого більшого цілого числа  
 RND- округлення даних до найближчого меншого цілого числа  
 RND округлення даних до найближчого цілого числа  
 TRUNC усікання дробової частини числа  
 INVI знаходження зворотного коду двійкового числа формату INT  
 INVD знаходження зворотного коду двійкового числа формату DINT  
 NEGI інвертування числа формату INT  
 NEG D інвертування числа формату DINT  
 NEGR інвертування числа формату REAL  
 ABS знаходження абсолютного значення числа формату REAL

#### Функції зрушення

SLW - побітове зрушення вліво вмісту молодшого слова акумулятора 1  
 SLD - побітове зрушення вліво вмісту акумулятора 1  
 SRW - побітове зрушення вправо вмісту молодшого слова ак. 1  
 SRD - побітове зрушення вправо вмісту акумулятора 1  
 SSI - побітове зрушення зі знаком вмісту молодшого слова ак. 1  
 SSD - побітове зрушення зі знаком вмісту акумулятора 1  
 RLD - циклічне зрушення вліво вмісту акумулятора 1  
 RRD - циклічне зрушення вправо вмісту акумулятора 1  
 - *n* на *n* позицій  
 - на число позицій, зазначене в акумуляторі Ассум 2  
 RLDA - циклічне зрушення вліво з використанням біта СС1  
 RRDA - циклічне зрушення вправо з використанням біта СС1

#### Логічні функції для слів даних

AW - операція І (AND) для слова даних  
 AD - операція І (AND) для подвійного слова даних  
 OW - операція АБО (OR) для слова даних  
 OD - операція АБО (OR) для подвійного слова даних  
 XOW - операція АБО виключаюче (Exclusive OR) для слова даних  
 XOD - операція АБО виключаюче (Exclusive OR) для подвійного слова даних  
 - *const* з константою формату слова даних або подвійного слова даних  
 - із вмістом акумулятора Ассум 2

## Функції керування в програмі

### Функції переходу

JU *мітка* безумовний перехід

Виконується перехід,

JC *мітка* якщо RLO = "1"

JCB *мітка* якщо RLO = "1" зі збереженням RLO

JCN *мітка* якщо RLO = "0"

JNB *мітка* якщо RLO = "0" зі збереженням RLO

JBI *мітка* якщо BR = "1"

JNBI *мітка* якщо BR = "0"

Виконується перехід,

JZ *мітка* якщо результат = "0"

JN *мітка* якщо результат < "0"

JP *мітка* якщо результат > "0"

JPZ *мітка* якщо результат >= "0"

JM *мітка* якщо результат < "0"

JMZ *мітка* якщо результат <= "0"

JUO *мітка* якщо результат некоректний

JO *мітка* перехід виконується при переповненні

JOS *мітка* перехід виконується при запомненному переповненні

JL *мітка* розподільник переходів

LOOP *мітка* циклічний перехід

### Функції обробки блоків

CALL FB виклик функціонального блоку

CALL FC виклик функції

CALL SFB виклик системного функціонального блоку

CALL SFC виклик системної функції

UC FB безумовний виклик функціонального блоку

CC FB виклик функціонального блоку за умовою

UC FC безумовний виклик функції

CC FC виклик функції за умовою

BEU безумовне завершення обробки блоку

BEC завершення обробки блоку за умовою

BE безумовне завершення обробки блоку

OPN DB виклик глобального блоку даних

OPN DI виклик екземплярного блоку даних

CDB обмін даними між регістрами блоку

L DBNO завантаження (load) номера глобального блоку даних

L DINO завантаження (load) номера екземплярного блоку даних

L DBLG завантаження (load) розміру глобального блоку даних

L DILG завантаження (load) розміру екземплярного блоку даних

NOP 0 нуль-операція

NOP 1 нуль-операція

BLD *n* інструкції відображення програми

ДОДАТОК Б. Варіанти завдань до роботи 3

№ Вар.	Система керування			Sm-Модулі автоматики				Характеристики двигунів			
	Кількість лінійних осей	Кільк. шпинделів	Панель оператора	Кіл. входів		Кіл. виходів		Подачі		Шпинделя	
				Дискр.	Аналог.	Дискр.	Аналог.	Момент Нм	Швидкість об/хв	Потужність кВт	Швидкість об/хв
1	3	2	OP 010	25x24V	4	15x24V	4	17	1500	12	3000
2	4	1	OP 015	30x24V	2	40x24V	2	20	3000	8	2000
3	2	1		20x24V		20x24V	2	6	1500	8	1500
4	2	2	OP 012	15x24V	3	30x24V	1	10	2000	30	1000
5	3	2	OP 010S	40x24V	2	15x24V	3	25	4000	15	2000
6	4	1	OP 015A	25x24V	4	15x24V	4	4	1500	20	1500
7	2	1	OP 010	30x24V	2	40x24V	2	12	3000	12	700
8	2	2	OP 015	20x24V		20x24V	2	30	1500	40	1000
9	3	1		15x24V	3	30x24V	1	25	2000	25	500
10	4	1	OP 012	40x24V	2	15x24V	3	17	1500	12	3000
11	2	1	OP 010S	25x24V	4	15x24V	4	20	3000	8	2000
12	2	2	OP 015A	30x24V	2	40x24V	2	6	1500	8	1500
13	3	2	OP 010	20x24V		20x24V	2	10	2000	30	1000
14	4	1	OP 015	15x24V	3	30x24V	1	25	4000	15	2000
15	2	1		40x24V	2	15x24V	3	4	1500	20	1500
16	2	2	OP 012	25x24V	4	15x24V	4	12	3000	12	700
17	3	2	OP 010S	30x24V	2	40x24V	2	30	1500	40	1000
18	4	1	OP 015A	20x24V		20x24V	2	25	2000	25	500
19	2	1	OP 010	15x24V	3	30x24V	1	6	1500	8	1500
20	2	2	OP 015	40x24V	2	15x24V	3	10	2000	30	1000
21	3	1		25x24V	4	15x24V	4	25	4000	15	2000
22	4	1	OP 012	30x24V	2	40x24V	2	4	1500	20	1500
23	2	1	OP 010S	20x24V		20x24V	2	12	3000	12	700
24	2	2	OP 015A	15x24V	3	30x24V	1	30	1500	40	1000
25	3	1		40x24V	2	15x24V	3	25	2000	25	500

### ДОДАТОК В. Варіанти завдань до роботи 4

№ Вар.	Система керування					Двигуни					
	Тип NCU	Панель оператора	Клавіатура CNC	Верстатний пульт	Модуль живлення	Подачи 2 шт (1FT6)			Шпинделя (1PH8)		
						Момент Нм	Швидкий. об/хв	Датчик	Потужність кВт	Швидкий. об/хв	Датчик
1	840D sl	OP 010C	KB 310C	MCP 483C	Smart	15	1500	AM2048S/R	5	1500	AM22DQ
2	840D sl	OP 012	KB 483C	MCP 310C	Active	24	2000	AM22DQ	20	700	AM2048S/R
3	840D sl	TP 015A	KB 310C	MCP 310C	Smart	2,5	3000	IC2048 S/R	8	2000	AM22DQ
4	840D sl	OP 015A	KB 483C	MCP 483C	Active	40	1500	IC22DQ	22	1000	IC2048 S/R
5	828D sl	вбудована	вбудована	MCP 310C	Combi	6	3000	AM22DQ	5	3000	AM2048S/R
6	828D sl	вбудована	вбудована	MCP 483C	Combi	12	3000	AM2048S/R	10	2000	AM22DQ
7	828D sl	вбудована	вбудована	MCP 310C	Combi	20	2000	AM22DQ	15	1000	IC2048 S/R
8	828D sl	вбудована	вбудована	MCP 483C	Combi	30	1500	IC2048 S/R	7,5	2000	IC22DQ
9	840D sl	TP 015A	KB 310C	MCP 310C	Smart	30	1500	AM2048S/R	12	1500	AM22DQ
10	840D sl	OP 015A	KB 483C	MCP 483C	Active	38	2000	AM22DQ	25	700	IC2048 S/R
11	840D sl	OP 010C	KB 310C	MCP 483C	Smart	18	3000	IC2048 S/R	18	2000	AM2048S/R
12	840D sl	OP 08T	KB 483C	MCP 310C	Active	45	1500	IC22DQ	22	1000	AM22DQ
13	828D sl	вбудована	вбудована	MCP 310C	Combi	10	2000	IC2048 S/R	10	1500	AM2048S/R
14	828D sl	вбудована	вбудована	MCP 483C	Combi	16	3000	IC22DQ	12	700	AM22DQ
15	828D sl	вбудована	вбудована	MCP 310C	Combi	20	2000	AM22DQ	14	2000	IC2048 S/R

16	828D sl	вбудована	вбудована	MCP 483C	Combi	25	1500	AM2048S/R	16	1000	IC2048 S/R
17	840D sl	OP 010C	KB 310C	MCP 310C	Smart	22	2000	AM22DQ	15	3000	IC22DQ
18	840D sl	OP 08T	KB 483C	MCP 483C	Active	36	3000	AM2048S/R	20	2000	IC2048 S/R
19	840D sl	OP 015A	KB 310C	MCP 310C	Smart	18	1500	AM22DQ	16	1000	IC22DQ
20	840D sl	TP 015A	KB 483C	MCP 483C	Active	42	3000	IC2048 S/R	24	2000	AM22DQ
21	828D sl	вбудована	вбудована	MCP 310C	Combi	8	2000	IC22DQ	8	1500	AM2048S/R
22	828D sl	вбудована	вбудована	MCP 483C	Combi	10	3000	AM22DQ	12	1000	AM22DQ
23	828D sl	вбудована	вбудована	MCP 310C	Combi	15	3000	AM2048S/R	10	3000	IC2048 S/R
24	828D sl	вбудована	вбудована	MCP 483C	Combi	18	2000	IC2048 S/R	15	2000	AM2048S/R