Розділ 6 CPU 21x-2BT02 з H1 / TCP/IP

Короткий огляд

В цьому розділі описується застосування процесорного модуля CPU 21x-2BT02 і організація зв'язку з використанням комунікаційних процедур H1 та TCP/IP. Тут також наведена методика конфігурування модулів за допомогою WinNCS з розглядом реального прикладу.

Зміст

Тема	Сторінка
Розділ 6 CPU 21x-2BT02 з H1 / TCP/IP	**6-1**
Базові положення	6-2
Планування мережі	6-7
MAC та IP-адреси	6-9
Розробка проекту для CPU 21x-2BT02	6-11
Приклад конфігурування CPU 21x-2BT02	6-23
Стадія запуску	6-34
Системні властивості CPU 21x-2BT02	6-35
Поєднання з іншими системами	6-37
Програма для тестування TCP/IP комунікації	6-40

Базові положення

Мережа

Мережа забезпечує зв'язок між різними станціями, дозволяючи їм обмінюватися даними один з одним.

Мережа складаються з PC, IPC, H1/TCP/IP-адаптерів і т.д.

Мережеві станції розділяються мінімальною відстанню і з'єднуються між собою за допомогою мережевого кабелю. Поєднання мережевих станцій і мережевого кабелю формує окремий сегмент, сукупність яких утворює Ethernet (фізика мережі).

Вита пара

На початку створення мережевих систем середовищем передачі даних в основному був коаксіальний кабель (жовтий кабель) або тонкий Ethernet-кабель (Cheapernet). Вони були витіснені кабелем вита пара, яка володіє більшою перешкодостійкістю. На борту модуля CPU 21xNET є роз'єм для підключення кабелю типу вита пара.

Вита пара складається з 8 скручених в пари провідників. Завдяки цьому він забезпечує підвищений рівень стійкості до електричних перешкод.

На відміну від шинних Ethernet мереж, що базуються на коаксіальному кабелю, мережі на основі витої пари утворюють топологічну модель типу точка-точка.

Найрозповсюдженішою моделлю локальних мереж являється зірка. За допомогою окремого кабелю кожна станція підключається до єдиного вузла - концентратора/комутатора, які забезпечують інтерфейс Ethernet.

Концентратор (репітер)

Концентратор є центральним елементом мережі Ethernet, реалізованої на базі витої пари.

Концентратор виконує функцію регенерації і підсилення сигналів в обох напрямках. У той же час, він повинен мати можливість виявляти і обробляти колізії мережевого сегменту. Концентратор не має своєї власної мережевої адреси, тому він не розпізнається системою як окремий пристрій.

В мережевій структурі концентратор з'єднується з пристроєм реалізації інтерфейсу Ethernet чи іншим концентратором або репітером.

Комутатор

Комутатор також є центральним елементом реалізації Ethernet на витій парі. Кілька станцій чи концентраторів з'єднуються між собою через комутатор, не впливаючи при цьому на загальну мережу. Інтелектуальне обладнання аналізує вхідні телеграми, які надходять на кожен порт комутатора і передає їх на кінцеву станцію уникаючи утворення колізій. Комутатор оптимізує пропускну здатність кожного підключеного сегменту мережі.

Контроль адреси

Ethernet підтримує принципи випадкового доступу до шини: кожна мережева станція може у будь-який час ініціювати передачу даних. Цей доступ координується системою CSMA/CD (Carrier Sense Multiple Access/Collision Detection): кожна станція "слухає" шинний кабель і отримує адресовані їй інформаційні повідомлення.

Станція ініціює передачу тільки коли лінія не зайнята. Якщо два учасника мережі одночасно починають передачу, вони виявляють це і зупиняють транзакцію. Перезапуск передачі ініціюється після закінчення випадкового часу затримки.

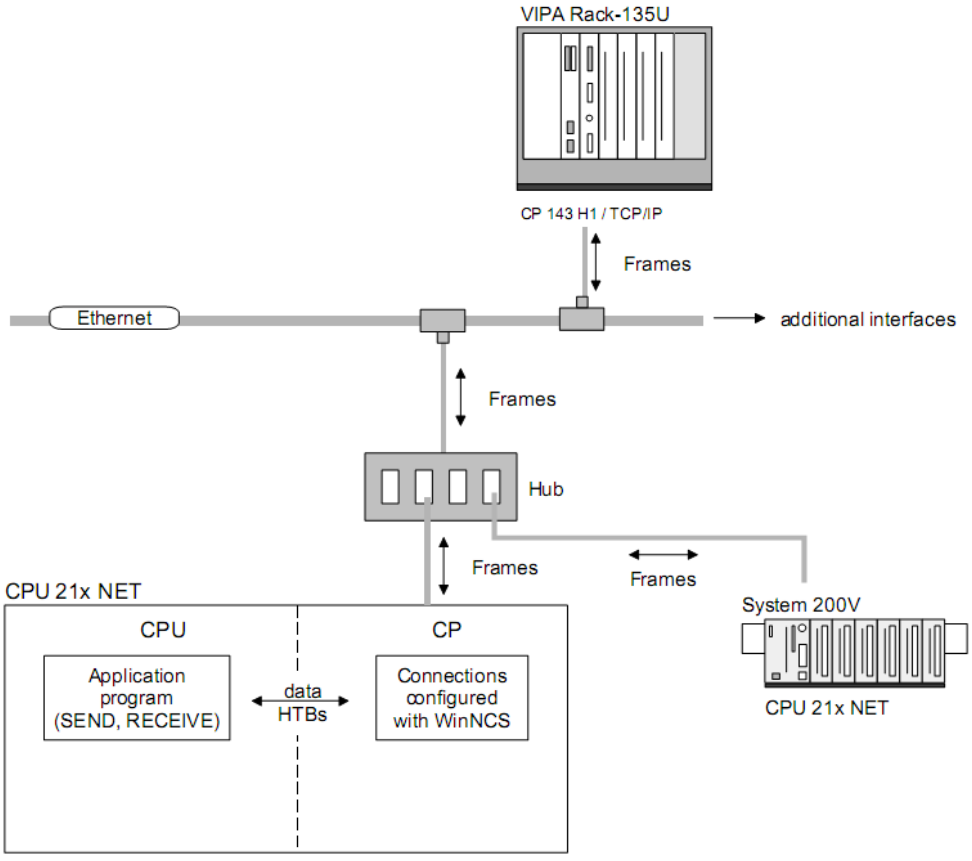
Комунікація

Внутрішній CP процесора 21x-2BT02 безпосередньо пов'язаний з CPU 21xNET за допомогою Dual-Port-RAM. Dual-Port-RAM ділиться на 4 рівні відрізки, які називаються кадрами.

Ці кадри доступні на CPU як стандартний інтерфейс CP. Обмін даними здійснюється за допомогою стандартних блоків обробники (SEND і RECEIVE).

ISO-транспорт (H1) і TCP/IP комунікація контролюється за допомогою зв'язків, які визначаються за допомогою конфігураційного інструменту VIPA WinNCS та безпосередньо переносяться в CPU через з'єднання по витій парі.

Більш детальну інформацію по конфігуруванню системи за допомогою WinNCS можна знайти в технічній документації (HB91).



ISO-транспорт (H1)

ISO-транспорт – це протокол, який базується на стандарті Ethernet.

Згідно з цим протоколом, обмін інформацією між станціями відбувається за допомогою ISO-транспортних кадрів, які передаються через комунікаційні зв'язки.

Транспортне сполучення представляє собою логічний зв'язок між двома точками доступу до сервісів передачі, сформоване на різних станціях. Передача використовує адресну інформацію, яка дає точний опис шляху між двома точками доступу.

Наступні параметри описують транспортне сполучення:

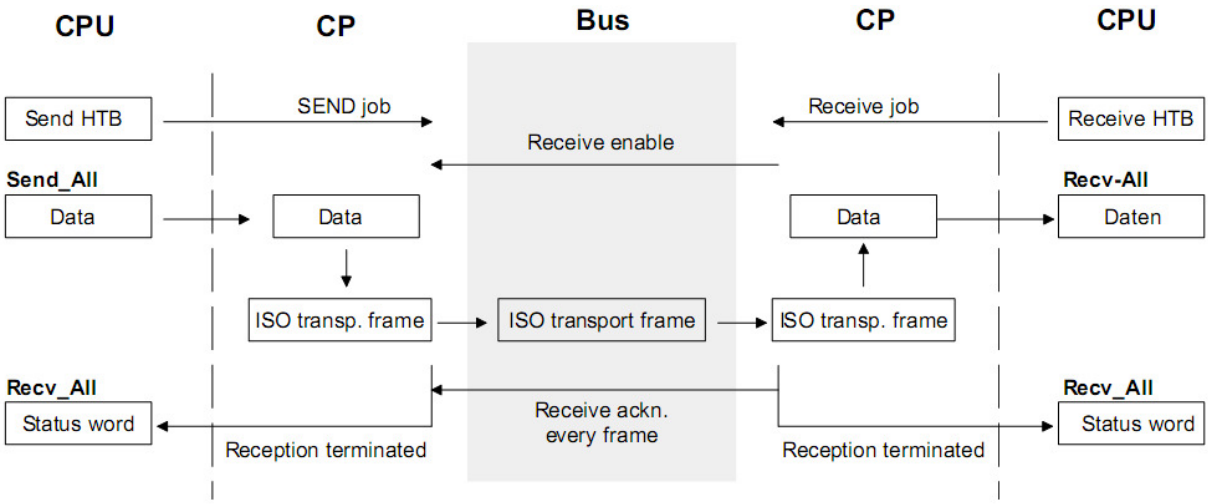
• *MAC-адреса*, також відома як адреса станції, дає унікальне визначення доступу до ресурсів станції.

• *TSAP Transport Service Access Points* визначає канали доступу для сервісів транспортного протоколу.

CP організовує буфер даних і за допомогою функції фонової комунікації SEND_ALL передає в нього інформацію. CP створює кадр ISO-транспорту і передає партнерській станції як тільки вона дає дозвіл на прийом. Коли партнерська станція отримала ISO-транспорті кадри, CP отримує підтвердження про отримання. Потім використовується функція фонового зв'язку RECEIVE_ALL, за допомогою якої здійснюється передача стану SEND у відповідні індикаційні слова.

Це гарантує організацію безпомилкового зв'язку.

Наступна діаграма графічно ілюструє комунікаційний принцип.



Примітка!

У зв'язку з великою кількістю підтверджувальних телеграм, які передаються ISO-транспортним зв'язком, навантаження на мережу значно вище, ніж в порівнянні з TCP/IP, однак при організації зв'язку по протоколу TCP/IP, надійність утвореного каналу комунікації нижча!

TCP/IP

TCP/IP протокол використовується при реалізації усіх великих систем для організації мережі між підрозділами.

З його допомогою вирішуються наступні завдання:

- централізоване управління і контроль промислових підприємств
- передача стану виробничого обладнання
- управління інформацією
- статистика виробництва
- передача великих обсягів даних.

TCP і IP являються протоколами необхідними для формування завершеної мережевої архітектури. Прикладний рівень забезпечує роботу таких додатків як "FTP" і "Telnet" для ПК.

Прикладний рівень CPU 21x2BT02 CP визначається програмою користувача з використанням стандартних блоків управління.

Користувацькі додатки для передачі даних використовують транспортний рівень з протоколами TCP і UDP, а також IP протокол для комунікації з інтернет-рівнем.

IP

Завданням IP являється передача пакетів даних від одного комп'ютера на інший. Це означає що IP виконує такі ж функції як конверт для листа. Адреса використовується в мережі для визначення призначення і маршруту пакетів даних.

Протокол ділить дані на невеликі порції, оскільки різні мережі використовують пакети даних різного розміру.

Кожному пакету присвоюється номер, який використовується для підтвердження прийому та змонтування вихідних даних. Щоб передати ці номерні послідовності через мережу, TCP і IP має унікальний конверт куди записуються ці номери.

TCP

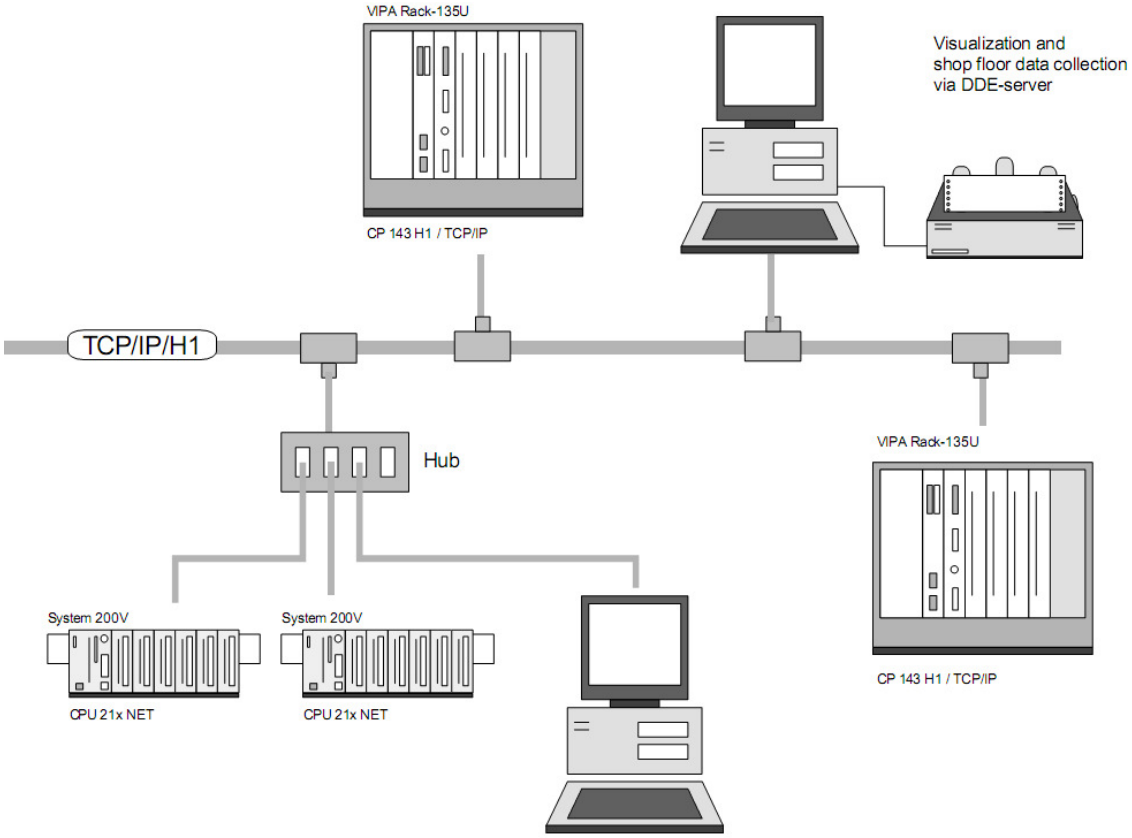
Пакет даних вставляється в конверт TCP. Потім він вставляється в

IP-конверт і передається в мережу. TCP забезпечує надійну передачу даних по мережі. TCP виявляє і виправляє комунікаційні помилки.

Таким чином, TCP-з'єднання є відносно безпечними.

UDP забезпечує більш швидкий канал зв'язку. Проте він не виявляє втрату пакетів даних та не перевіряє їх послідовність. UDP не є надійним протоколом.

Приклад ISO-транспорту або TCP/IP мережі



Планування мережі

Стандарти та принципи побудови

Для реалізації надійного зв'язку між різними станціями необхідно дотримуватись відповідних правил та положень. В стандартах визначається протокол даних, спосіб доступу до шини та інші принципи, важливі для комунікації.

VIPA CPU 21xNET розроблено відповідно до стандартів, визначених в ISO.

Міжнародні та національні комітети визначили такі стандарти і керівні принципи для мережевих технологій:

ANSI Американський національний інститут стандартів

Стандарт ANSI X3T9.5 в даний час визначає положення для високошвидкісних локальних мереж (100 МБ/с) на основі оптично-волоконних технологій.

(FDDI) Волоконний інтерфейс передачі даних.

CCITT Консультативний комітет міжнародного телефонного і телеграфно зв'язку

Цей консультативний комітет підготував положення підключення промислових мереж (MAP) до офісних мереж (TOP) на базі Глобальних мереж (WAN).

ECMA Європейська асоціація виробників комп'ютерів

Розробила різні MAP і TOP стандарти.

EIA Електротехнічна промислова асоціація (США)

Цей комітет визначив стандарти RS-232 (V.24) і RS-511.

IEC Міжнародна електротехнічна комісія.

Визначила деякі спеціальні стандарти, наприклад, використання польової шини.

ISO Міжнародна організація зі стандартизації.

Ця асоціація національних організацій по стандартизації розробила OSI-модель (ISO/TC97/SC16). Вона займається визначенням стандартів передачі даних. Стандарти ISO включені в різні національні стандарти, такі як, наприклад, UL і DIN.

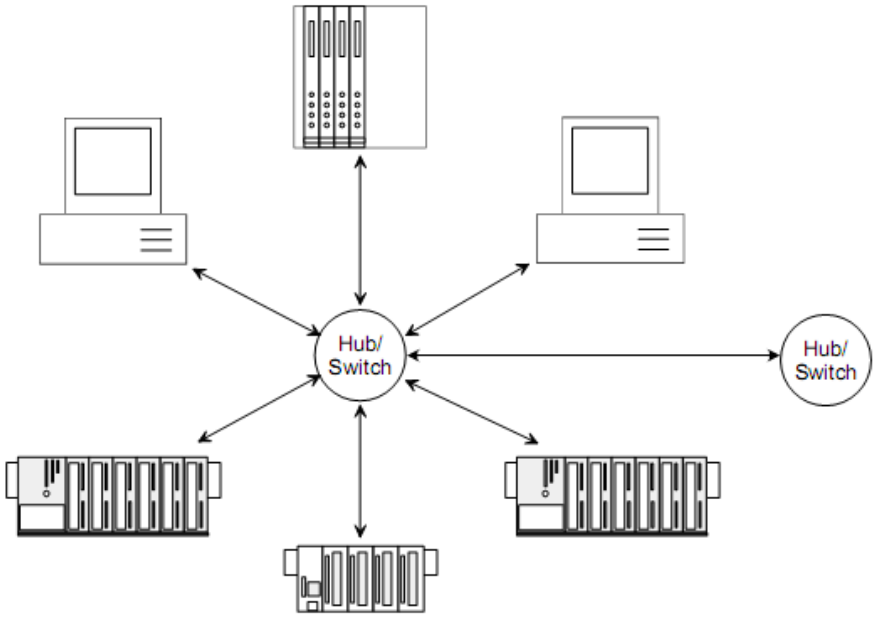
IEEE Інститут інженерів з електротехніки та електроніки (США).

В рамках проекту група 802 визначила LAN-стандарти швидкості передачі даних від 1 до 1000 МБ/с. IEEE стандарти часто є основою для ISO-стандартів, наприклад, IEEE 802.3 = 8802.3 ISO.

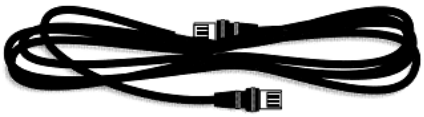
Огляд компонентів

CPU 21x-2BT02 використовується виключно для роботи в мережі, яка побудова на витій парі.

В такій мережі усі станції підключаються по топології зірка до концентратора/комутатора, який в свою чергу, також може підключатися до іншого концентратора/комутатора. Дві з'єднані станції утворюють сегмент, в якому відстань між пристроями не повинна перевищувати 100м.



Кабель вита пара:



Має 8 провідників скручених в пари.

Провідники мають діаметр від 0,4 до 0,6мм.

Аналітичні викладки

• Розмір області, яка повинна охоплюватися мережею?

• Скільки сегментів забезпечать найкраще рішення для фізичного прокладання мережі (простір, перешкоди)?

• Скільки мережевих пристроїв (SPS, IPC, PC передавач, мости) повинні бути підключені до кабелю?

• Яка відстань між різними станціями в мережі?

• Орієнтоване розширення системи, очікувана кількість підключень?

• Який обсяг даних повинен бути оброблений (ширина смуги пропускання, час доступу, сек.)?

Креслення мережевої діаграми

Намалюйте схему мережі. Визначте кожен апаратний елемент (тобто станції, кабель, концентратор, комутатор). Дотримуйтесь заданих правил і обмежень.

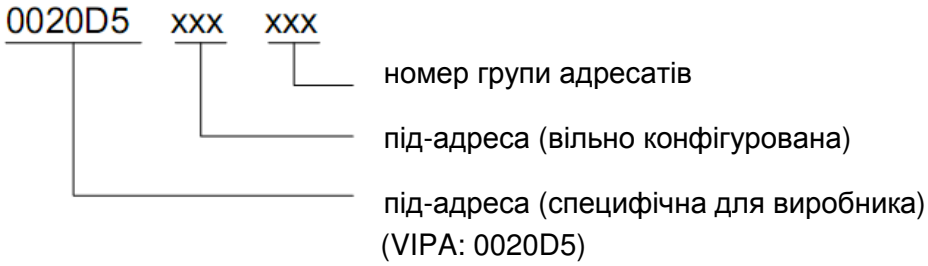
Виміряйте відстань між усіма компонентами системи, щоб не перевищити максимально допустиму відстань між ними.

MAC та IP-адреси

Структура Ethernet-apec

До станції можна звертатися за допомогою MAC-адреси, яка також відома як адреса станції. Усі MAC-адреси станцій в мережі повинні бути унікальним.

MAC-адреса складається з наступних елементів:



MAC-адреса має довжину 6Байт. Перші тири байти визначаються компанією-виробником. Вони присвоюються IEEE-комітетом. Останні три байти визначаються за необхідністю.

Адміністратор мережі визначає MAC-адресу

Широкомовна адреса (для передачі повідомлень на всі станції мережі) має наступний вигляд:

FFFFFFFFFFFFh

Структура IP-адреси

IP-адреса – 32Біта, яка повинна бути унікальною в межах мережі і складається з 4 чисел, розділених крапкою.

Кожна IP-адреса має наступний вигляд: **XXX.XXX.XXX.XXX**

Діапазон: 000.000.000.000 до 255.255.255.255

Адміністратор мережі також визначає IP-адреси.

Широкомовна адреса (передає повідомлення для всіх станцій) завжди:

255.255.255.255

Увага!

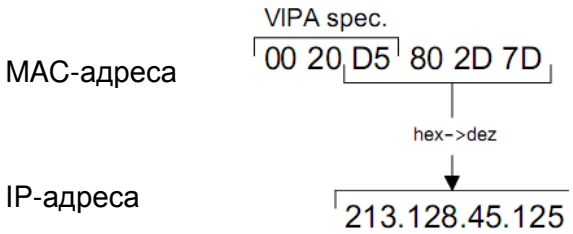
На деякі IP-адреси існують обмеження, оскільки вони зарезервовані для спеціальних служб!

Початкова адреса

Коли CPU 21x-2BT02 вмикається перший раз, він має наперед визначену початкову MAC-адресу (адреса станції).

Її можна зчитати з маркування нанесеного на бічну сторону модуля.

Ця MAC-адреса використовується тільки при першому ввімкненні модуля. У відповідності з нею CPU має унікальну IP-адресу, яка вираховується наступним чином:



Примітка!

Взаємозв'язок між MAC- і IP-адресами існує тільки при першому ввімкненні.

Для присвоєння іншої адреси Ви завжди можете скористатися функцією CP-Init в WinNCS.

Увага!

Після зміни оригінальна Ethernet адреса не може бути відновлена, оскільки виконання загального скидання для секції CP не можливе.

Розробка проекту для CPU 21x-2BT02

Вступ

Процедура розробки проекту для TCP/IP складається з трьох частин:

- **CP-конфігурування** за допомогою VIPA WinNCS (Ethernet з'єднання)
- **Апаратне конфігурування** (включення CP в CPU)
- **Програмування PLC** шляхом заливки прикладної програми користувача (PLC з'єднання).

Коротке знайомство

Передумови

CP ввімкнено і підключено до Ethernet.

Послідовність кроків по розробці CP-проекту в WinNCS



- Запустіть WinNCS і встановіть вибравши кнопку функціональність "Ethernet".
- Створіть проект за допомогою меню "Ethernet" **File** > *Project set-up/open.*
- Натиснувши на кнопку 

переведіть систему в онлайн режим.
- За допомогою MAC-адреси (див. маркування модуля) можна доступитись до CP. Для цього для ISO-транспорту натисніть кнопку "H1 protocol" 

і в введіть MAC-адресу в якості адреси станції.
- Встановіть з'єднання за допомогою 

.
- Кнопкою 

встановіть CP в режим Software STOP.
- Тепер Ви можете імпортувати конфігурацію CP в WinNCS, адаптувати і завантажити її назад в CP або задати властивості станції "CP-Init" і налаштувати відповідні з'єднання.

Передача CP проекту

У вікні "network" виберіть станцію, куди буде передано конфігурацію.

• Почніть передачу в CP за допомогою 

. Якщо з'явиться запит на NCS файл, то це означає що у вікні "network" не вказано модель CP! Поверніться до попереднього кроку, виберіть відповідний об'єкт та знову почніть передачу.

• Оскільки дані зберігаються в оперативній пам'яті, через 

їх необхідно перенести у Flash ROM-пам'ять. При цьому обновиться дата виконання команди.

• Після закінчення передачі перезавантажте CP через 

.

Тепер CP доступний через присвоєні адресні параметри.

Апаратна конфігурація

Умова: SIMATIC Manager Siemens, починаючи з V. 5.1 і SIMATIC NET

• Запустіть Simatic Manager і створіть новий проект.

• Додайте нову станцію System300 через меню **Insert** > *Station* > *SIMATIC 300-Station*.

• Активуйте "SIMATIC300" і натиснувши кнопку "Hardware" відкрийте апаратний конфігуратор.

• Зконфігуруйте шасі (Simatic300 > Rack-300 > Profile rail).

• CPU 21xNET зконфігуруйте як Siemens CPU 315-2DP артикул 6ES7 315-2AF03-0AB0 з апаратної директорії Simatic300 > CPU-300 > CPU 315-2 DP.

• При необхідності зконфігуруйте CPU та модулі. Вікно параметризації з'явиться при подвійному натисненні на відповідному модулі.

• До підмережі додайте System "VIPA_CPU21x". Необхідні елементи Ви знайдете в директорії PROFIBUS DP > Additional Field Devices > IO > VIPA_System_200V. Призначте цій слейв-станції PROFIBUS-адресу 1.

• Встановіть VIPA CPU 21x-2BT02 в 1$^{-й}$ слот в конфігураторі.

• Послідовно додайте до System 200V периферійні модулі, починаючи з позиції 1.

• Збережіть проект.

Передача апаратної конфігурації відбувається разом з програмою користувача.

Програма користувача

• Створіть канал зв'язку між процесором і CP за допомогою блоку SYNCHRON.

• Запрограмуйте відповідні блоки SEND та RECEIVE для ініціалізації відправки і отримання команд.

• Запрограмуйте блоки SEND_ALL та RECEIVE_ALL для передачі даних.

Перенесення користувацьких додатків і конфігурації обладнання

Для передачі програми та конфігурації обладнання існує декілька можливостей:

а) передача через MPI

б) передача через MMC

Передача через MPI

• Підключіть PG або PC через MPI до CPU. Для організації послідовного з'єднання типу точка-точки Ви можете використати VIPA Green Cable (докладніше див. документацію "CPU 21x", розділ "Передача проекту").

• Зконфігуруйте MPI інтерфейс комп'ютера в Siemens SIMATIC Manager через меню **Options** > *Set PG/PC Interface*. При використанні кабелю Green Cable встановіть швидкість передачі даних 38400Baud.

• Через меню **PLC** > *Download* передайте проект в CPU.

Передача через MMC

Читання з MMC завжди відбувається після виконання операції OVERALL_RESET.

Для запису конфігурації на MMC необхідно використати **PLC** > *Copy RAM to ROM* або зчитувальний пристрій MMC від VIPA (номер для замовлення: VIPA 950-0AD00).

Будь ласка, подбайте про те, щоб Ваш проект зберігався в кореневому каталозі і мав ім'я файлу: **S7PROG.WLD**.

Під час процесу запису буде блимати жовтий "MMC"-LED індикатор CPU.

В цей час відбувається запис на внутрішню флеш-пам’ять процесора.

Після успішного запису в діагностичному буфері з’явиться строка 0xE200 і 0xE300.

Якщо на MMC записано не коректний додаток або якщо передача конфігурації відбулась невдало, автоматично виконається процедура OVERALL-RESET процесора і STOP-LED блимне три рази.

На цьому короткий вступ закінчено. Нижче наведено докладний опис процесу створення проекту.

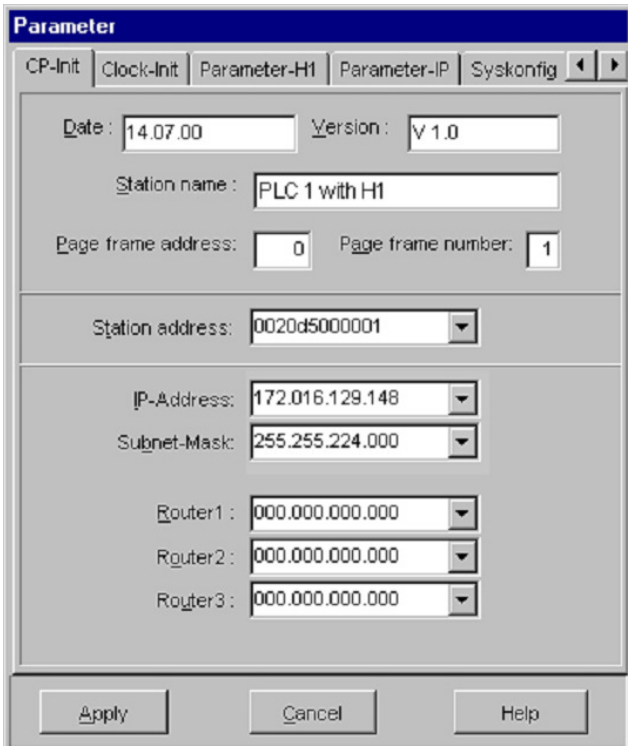
Конфігурування CP за допомогою WinNCS

CP секція CPU 21x-2BT02 може бути зконфігурована лише за допомогою WinNCS від VIPA. Цей процес складається з трьох етапів:

- Базове конфігурування CP
- Конфігурування комунікаційних модулів
- Передача даних конфігурації в CP.

Початкове конфігурування CP

Тут визначається адреса та інші параметри ідентифікації станції. Через розділ "Ethernet" у вікні *network* встановіть нову станцію і введіть необхідні дані конфігурації.



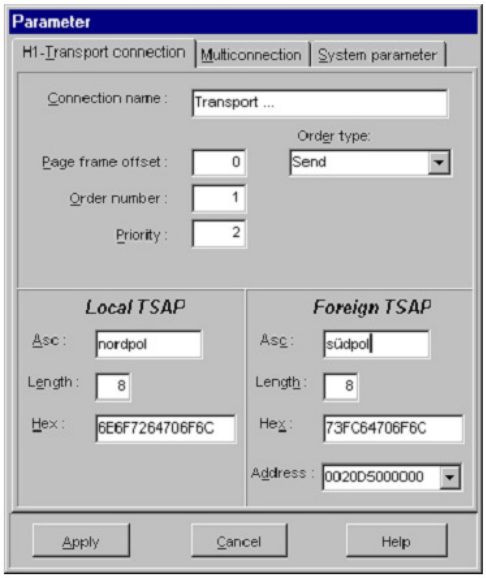
Основні параметри конфігурації CP визначають поведінку станції в мережі.

Конфігурування комунікаційних блоків

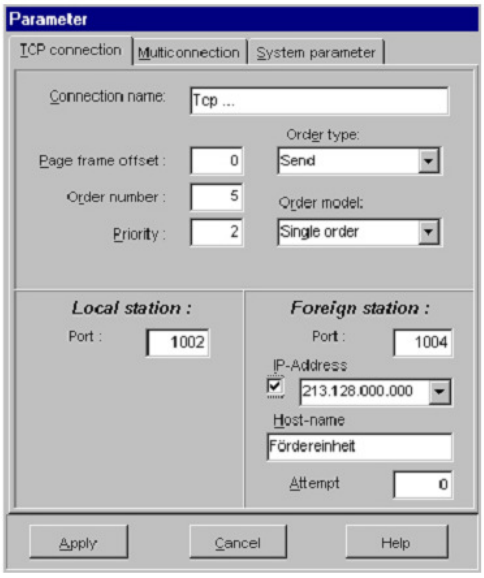
Комунікаційний блок містить віддалені параметри, тобто параметри, які орієнтовані на обслуговування партнерів по мережі, і локальні параметри, тобто параметри, які відносяться до програми PLC.

В залежності від обраного протоколу, Ви можете налаштувати ISO-транспорт (H1) або TCP/IP з'єднання. Для цього необхідно вибрати символ станції і вставити / зконфігурувати відповідне з'єднання.

ISO transport (H1) connection



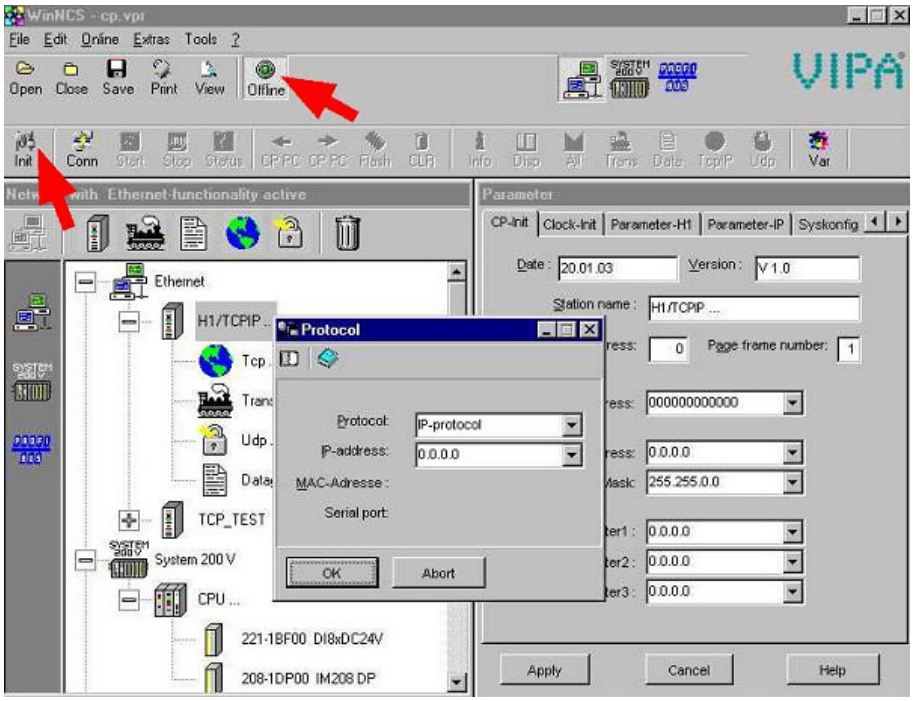
TCP/IP connection



Передача конфігурації в CP

Після закінчення конфігурування усіх з'єднань, необхідно передати ці параметри в CP. Для цього служить пункт меню "Module transfer functions" в WinNCS.

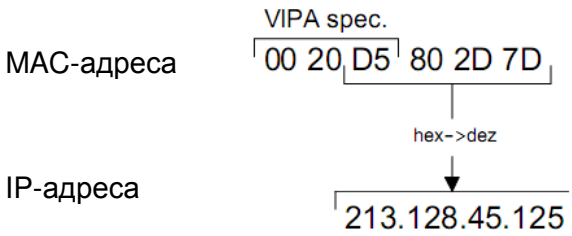
Передача конфігурації почнеться після активації функції „online functions“ і натиснення кнопки INIT:



Як описано вище, CPU 21x-2BT02 поставляється з передустановленою MAC-адресою.

Її можна знайти на маркуванні нанесеному на бічній стінці корпусу модуля.

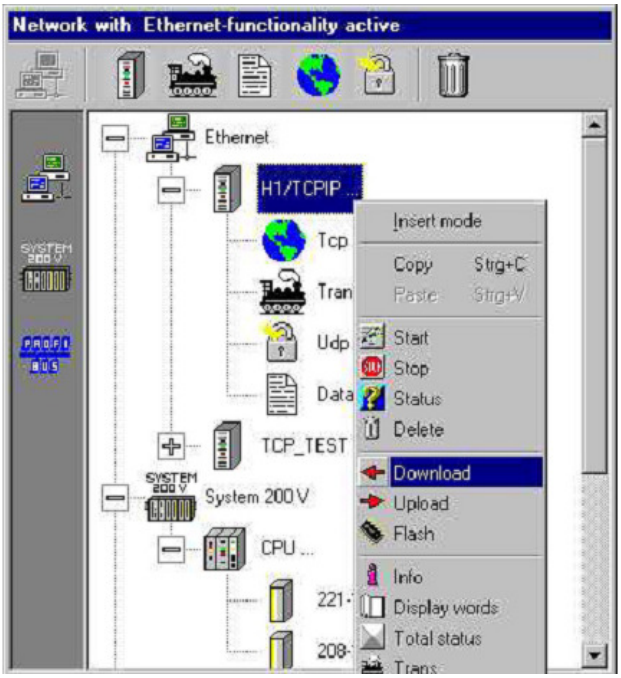
Ця MAC-адреса використовується тільки при першому ввімкненні модуля. Потім вона змінюється на індивідуальну IP-адресу, яка вираховується за наступними правилами:



У вікні *Protocol* виберіть „IP protocol“ і задайте необхідну IP-адресу. Підтвердіть введення кнопкою [OK].

Тепер перейдіть у вікно *network* та активуйте відповідну станцію. В контекстному меню, яке викликається правою кнопкою миші, виберіть пункт „Download“.

Після цього Ваш проект буде переданий безпосередньо в RAM вибраного CP.



Оскільки дані поки що зберігаються в незахищеній пам'яті, за допомогою кнопки 

Ви повинні зберігати їх у Flash ROM. І так необхідно повторювати при кожному новому поновленні конфігурації.

Програмування додатку PLC

Для того, щоб PLC міг обробляти комунікаційні запити, в CPU необхідно завантажити відповідну програму, яка містить блоки обробки (SEND, RECEIVE, ...).

Також необхідно, щоб за допомогою програми PLC було організовано канал зв'язку між CPU та CP ("synchronization"). Ця задача виконується за допомогою блоку SYNCHRON.

Передача і прийом виконується за допомогою блоків SEND та RECEIVE.

Передача даних ініціюється за допомогою команд SEND_ALL та RECEIVE_ALL.

Повідомлення про помилку відображається в індикаторному слові.

Синхронізація

Використання інтерфейсу CP на етапі ініціації вимагає синхронізації OB 100 з CPU. Для цього використовується блок обробки SYNCHRON.

Після включення живлення CPU21x-2BT02 потребує 15 секунд для коректного проходження процедури завантаження. Якщо протягом цього часу PLC повинен згенерувати запит на синхронізацію, в діагностичний буфер повернеться помилка конфігураційного байту PAFE. Це повідомлення видаляється після завершення процедури ініціації CP.

Таймер в цьому блоці на початковому етапі встановлюється в 20 сек. Обробка буде зупинена, якщо протягом цього періоду синхронізація завершиться не належним чином.

Розміри блоку

У наступній таблиці наведено доступні розміри блоку.

Розмір блоку	Розмір блоку CP в байтах
0	За-замовчуванням
1	16
2	32
3	64
4	128
5	256
6	512
255	512

Цикл

Блоки передачі і прийому SEND та RECEIVE, які відповідають за ініціацію зв’язку, повинні бути налаштовані в програмному циклі OB1.

Блоки SEND_ALL і RECEIVE_ALL виконують передачу даних.

Для реалізації чисто пасивного з'єднання необхідні лише компоненти SEND_ALL або RECEIVE_ALL.

Захист передачі даних здійснюється за рахунок інтегрування різних контрольних пунктів, на яких здійснюється оцінка показників слова.

Блоки обробки

У наступній таблиці перераховані необхідні блоки обробки. Більш детальну інформацію можна знайти в технічній документації CPU 21x – Список операцій.

SFC	Маркування	Опис
SFC 228	RW_frame	Читання\запис образу сторінки
SFC 230	Send	Відправка в CP через образ сторінки
SFC 231	Receive	Отримання з CP через образ сторінки
SFC 232	Fetch	Вибірка починає запрос даних за допомогою образу сторінки. FETCH допускається тільки з ідентифікатором RW і забезпечує порядок ініціалізації для читання
SFC 233	Control	Блок CONTROL призначений для запиту статусу і забезпечує оновлення порядку в ANZW
SFC 234	Reset	Блок RESET ініціалізує очищення порядку визначеного з'єднання
SFC 235	Synchron	Блок SYNCHRON служить для синхронізації CPU і CP під час запуску. У той же час, образ сторінки очищається і розмір блоку між CPU і CP розраховується повторно. Активний обмін даними може виконуватись тільки між синхронізованими сторінками образів.
SFC 236	Send_All	Ініціалізація передачі даних від CPU до CP
SFC 237	Recv_All	Ініціалізація прийому даних від CP до CPU
SFC 238	Control1	Контроль комунікації образу сторінки типу ANZW: IND покажчика і параметрів

Передача програми і апаратної конфігурації

Для передачі програми користувача і апаратної конфігурації існують такі можливості:

а) передача через MPI

б) передача через MMC

Передача через MPI

• Підключіть PG або PC через MPI до CPU. Якщо програмний пристрій не має MPI-інтерфейсу, Ви можете організувати послідовне з'єднання типу точка-точки використавши VIPA Green Cable.

"Green Cable" має код для замовлення VIPA 950-0KB00 і може застосовуватись лише для CPU з MP^{2}I-інтерфейсом. Будь ласка, ознайомтесь з порадами щодо цього в розділі "Основи".

• Зконфігуруйте MPI інтерфейс комп'ютера.

• Через меню **PLC** > *Download* передайте проект в CPU.

• Для додаткового збереження проекту на MMC необхідно вибрити меню **PLC** > *Copy RAM to ROM*.

Під час процесу запису "MC"-LED індикатор на CPU буде блимати. Успішний запис реєструється в системному буфері. Процес запису повністю закінчиться, коли LED перестане світитися.

Поради по конфігуруванню інтерфейсу MMC можна знайти в технічній документації на програмне забезпечення.

Далі описується використання VIPA "Green Cable" в комбінації з програмним інструментом від Siemens.

"Green Cable" встановлює послідовне з'єднання між COM-інтерфейсом PC і MP^{2}I інтерфейсом CPU.

Увага!

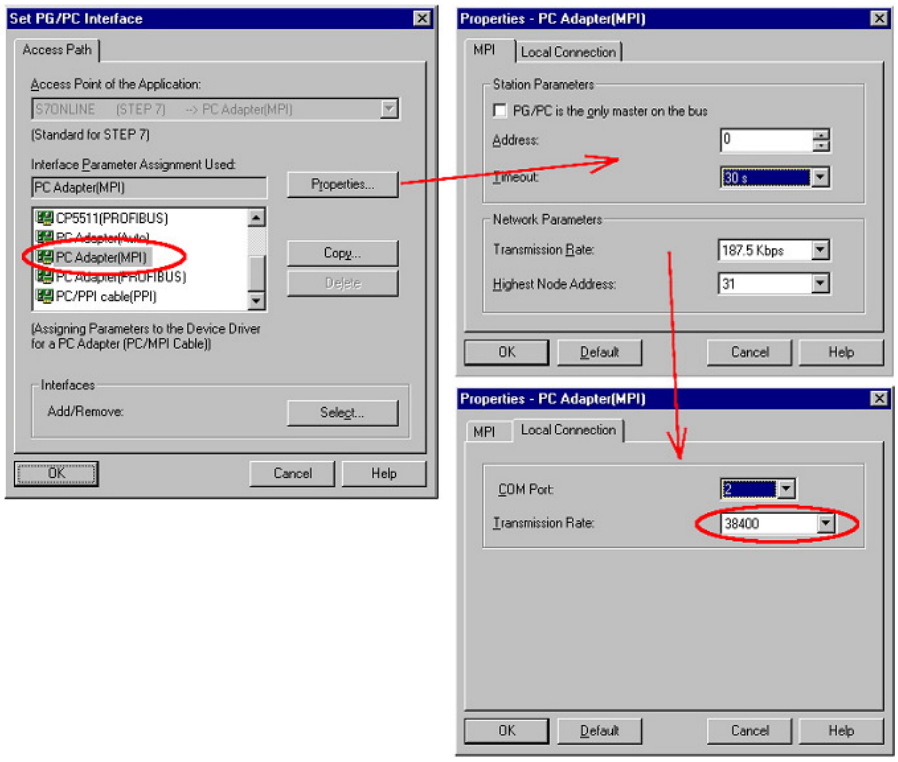
Майте на увазі, що використання "Green Cable" допускається тільки для підключення до MP^{2}I інтерфейсу VIPA CPU.

Послідовність дій

• Запустіть Simatic Manager від Siemens.

• Виберіть **Options** > *Set PG/PC Interface*.

→ відкриється наступне діалогове вікно в якому Ви зможете налаштувати MPI інтерфейс:



• Зі списку виберіть "PC Adapter (MPI)". Якщо він там відсутній, необхідно його додати уручну.

• Натисніть кнопку [Properties].

→ після цього зконфігуруйте наступні два вікна, як зображено на малюнку.

Примітка!

При використанні Green Cable установіть швидкість передачі даних в 38400 Baud.

Передача через MMC

В якості зовнішнього носія інформації можна використовувати MultiMediaCard (MMC) VIPA 953-0KX10.

Читання з MMC завжди відбувається після OVERALL_RESET.

Для запису конфігурації на MMC необхідно виконати команду запису в апаратному кофігураторі або використати зчитувальний пристрій MMC від VIPA (номер для замовлення: VIPA 950-0AD00). Карточки MMC від VIPA мають файлову структуру FAT16.

Необхідні файли

Ви можете зберігати кілька проектів і каталогів на MMC.

Будь ласка, переконайтесь що проект, який ви хочете передати, зберігається в кореневому каталозі і має ім'я файлу: **S7PROG.WLD**.

Передача CPU → MMC

Підключіть MMC і виконайте команду передачі вмісту буферу **PLC** > *Copy RAM to ROM*.

Під час процесу запису буде блимати жовтий "MMC"-LED індикатор CPU.

В цей час відбувається запис на внутрішню флеш-пам’ять процесора.

Контроль процесу передачі

Після передачі в діагностичному буфері CPU з’явиться відповідний запис ID. Для перегляду цього буфера в SIMATIC Manager необхідно вибрати **PLC** > *Module Information*. Відкривши закладку "Diagnostic buffer" Ви можете переглянути вікно діагностики. Більш докладно дивіться в технічній документації в розділі "Перелік інструкцій".

При записі на MMC можуть виникати наступні події:

ID події	Опис
0xE100	Помилка доступу до MMC
0xE101	Помилка файлової системи MMC
0xE102	Помилка MMC FAT
0xE200	Закінчення запису на MMC
0xE300	Закінчення запису на внутрішню флеш пам’ять

Якщо на MMC записано не коректний додаток або якщо передача конфігурації відбулась невдало, автоматично виконається OVERALL-RESET процесора і STOP-LED блимне три рази.



Примітка!

При успішному запису в діагностичний буфер занесеться ідентифікатор 0xE200 і 0xE300.

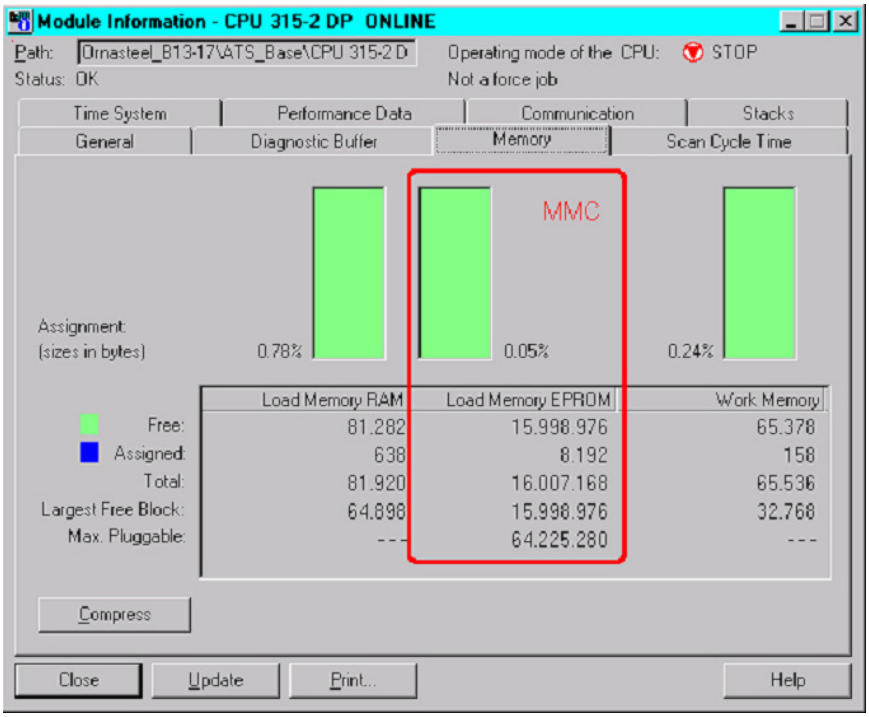
Передача MMC → CPU

Передача користувацької програми з MMC в CPU завжди відбувається після виконання OVERALL_RESET. Під час процесу запису буде блимати жовтий "MC" LED процесора.

Якщо на підключеній карточці MMC відсутня коректна програма, процес передачі перерветься і CPU виконає операцію OVERALL_RESET. При цьому STOP-LED індикатор блимне тричі.

Моніторинг розміру пам'яті MMC

Для контролю пам'яті MMC в SIMATIC Manager необхідно вибрати **PLC** > *Module Information*. Відкривши закладку "Memory", Ви отримаєте доступ інформації про поточне використання пам’яті CPU.



Доступ до внутрішньої флеш-пам’яті

Після виконання описаних вище операцій, вміст RAM буфера батареї передається в MMC і у внутрішню флеш-пам’ять.

Команда запису виконується в апаратному конфігураторі Siemens через меню **PLC** > *Copy RAM to ROM*.

Доступ на читання з внутрішньої флеш-пам’яті відбувається тільки при порожньому буфері і відсутній MMC-карточці.

Примітка!

Якщо програма користувача більша за доступну пам’ять CPU, передача вмісту MMC не відбудеться.

Перед передачею виконайте стиснення файлу, оскільки ця операція не здійснюється автоматично.

Керування процесом

• Кнопкою 

в WinNCS переведіть систему в онлайн режим.

• Виберіть "IP protocol" 

і введіть нову IP-адресу.

• Встановіть з'єднання за допомогою 

. Тепер Ви можете обмінюватися даними через адресу, вказану в CP-Init.

• CP повинен бути в RUN. Перевірте це через 

. Якщо CP в *Idle*-режимі, синхронізація з CPU не відбулася. Будь ласка, контролюйте SYNCHRON в блоці OB 100.

• Для контролю за системою Ви маєте можливість слідкувати за зведеною інформацією про стан TCP з'єднання через 

.

На цьому конфігурування CPU і CP закінчено.

Приклад конфігурування CPU 21x-2BT02

Вступ

Цей розділ містить базову інформацію стосовно використання мережі TCP/IP для System 200V. Метою є створення невеликої комунікаційної системи між двома VIPA CPU 21x-2BT02 та демонстрація методики контролю мережевих процесів.

Вимоги

Необхідні знання блоків обробки CP. Блоки обробки CP – це стандартні функціональні блоки. Вони забезпечують можливість використання комунікаційних функцій в додатках програмованих логічних контролерів.

Мінімальна технічне обладнання, необхідне для реалізації розглянутого прикладу:

Обладнання

- 2 CPU 21x-2BT02 від VIPA
- 1 PC або PG з інтерфейсом Ethernet

Комунікаційні лінії

- 3 мережевих кабелі
- 1 міні-свіч CM 240

Пакет програмного забезпечення

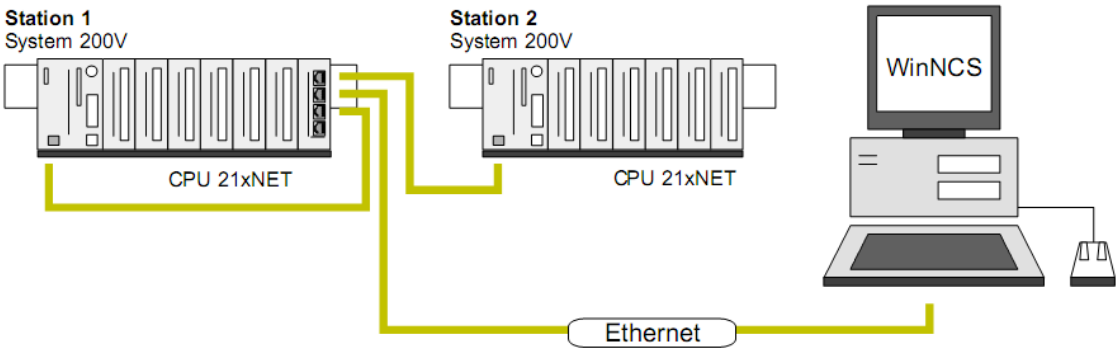
- Конфігураційна програма WinNCS від VIPA
- WinPLC7 від VIPA або Siemens SIMATIC Manager для CPU 21xNET

При реалізації прикладу буде зконфігуровано два CPU та комунікаційний процесор з за допомогою інструменту WinNCS.

Примітка!

Цей приклад можна закачати з нашого порталу: ftp.vipa.de/support/demofiles. Програма напряму завантажується безпосередньо в обидва CPU.

Структура



Задачі

Наведений приклад ілюструє користувацьку програму, яка базується на виконанні комунікаційних задач.

Обидва CPU містять однакову програму PLC, але конфігурація CP в них повинна бути різна.

Станції відправляють і отримують 16 слів даних в секунду.

• Блок даних DB11 передає слова даних від DW0 до DW15 з інтервалом 1 сек. Слово даних DW0 в DB 11 використовується як лічильник повідомлень. Він збільшується тільки якщо попередня команда передачі була оброблена правильно (завершена без помилок). Решта слів даних (від DW1 до DBB 32) можуть бути використані для передачі користувацьких даних.

• Приймаюча станція зберігає дані в DB12 (від DBB 0 до DW15).

• SEND конфігурується з номером A-No. = 1 і зі зміщенням образу сторінки SSNR = 0.

• RECEIVE конфігурується з номером A-No. = 11 і зі зміщенням образу сторінки SSNR = 0.

• Повинні бути налаштовані параметри джерела і призначення даних.

Далі наводяться додаткові відомості про конфігурацію блоків обробки в CP і методологія реалізації ISO-транспорту (H1) та TCP/IP.

Конфігурування в WinNCS

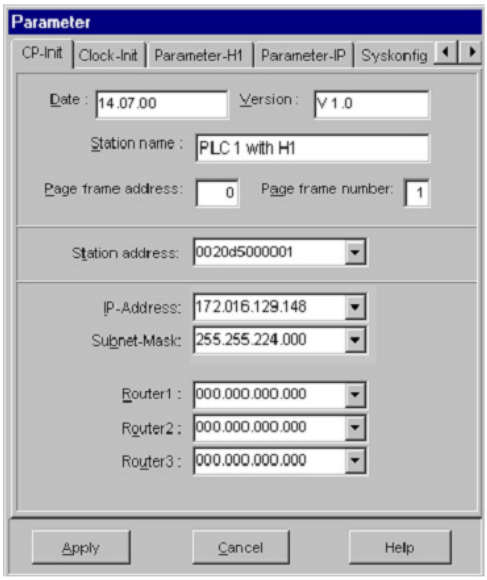
Два CP конфігуруються виключно за допомогою WinNCS. Запустіть WinNCS і створіть проект, який містить функції групи "Ethernet_H1". Процедура однакова для обох станцій і складається з трьох етапів:

- Базове конфігурування CP
- Конфігурування комунікаційних блоків
- Передача даних конфігурації в CP.

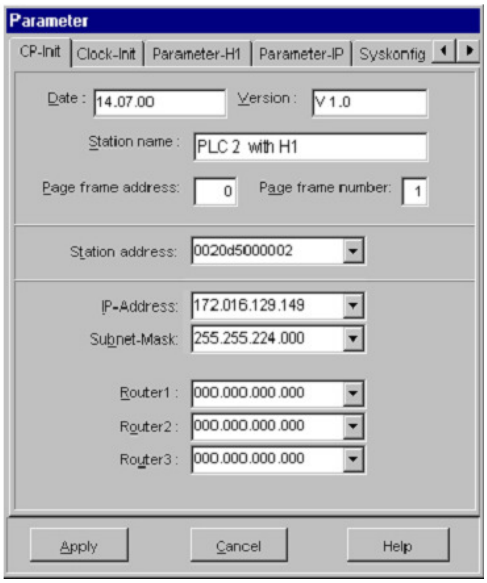
Базове конфігурування CP

Розмістіть дві станції і встановіть наступні налаштування:

Станція 1



Станція 2



Необхідні адреси станцій отримайте у Вашого системного адміністратора.

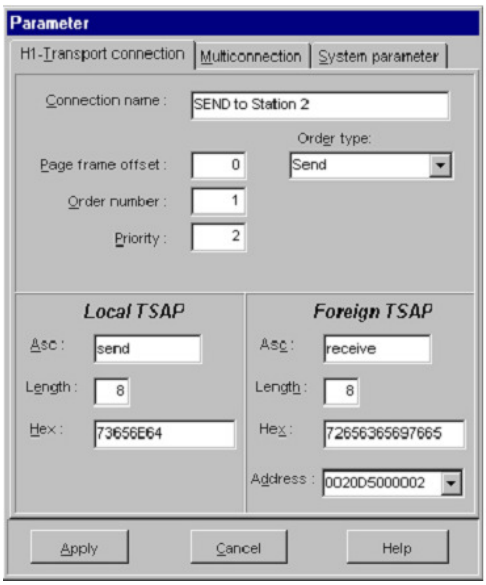
При необхідності в конфігураційні вікна введіть додаткові параметри.

Конфігурування зв'язку

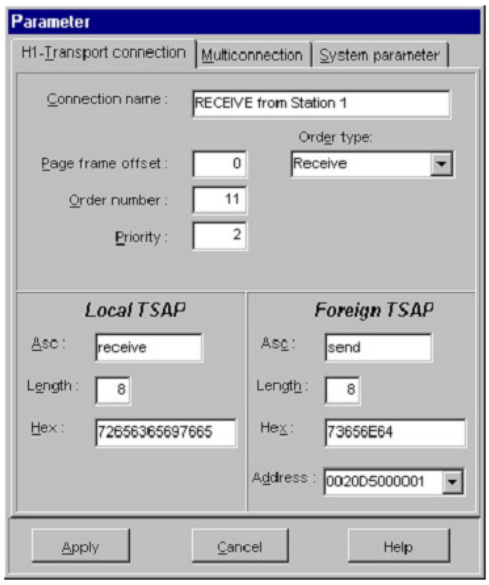
ISO-транспорт (H1)

Конфігурування комунікації ISO-транспорт виконується натисненням кнопки  і конфігуруванням відповідних вікон параметризації.

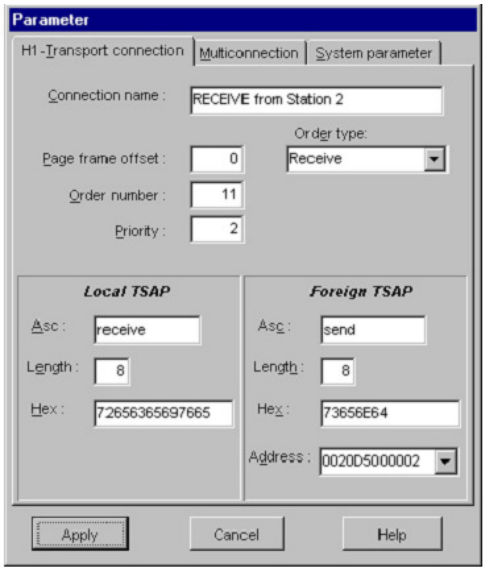
Станція 1
Відправка на адр.: 0020D5000002



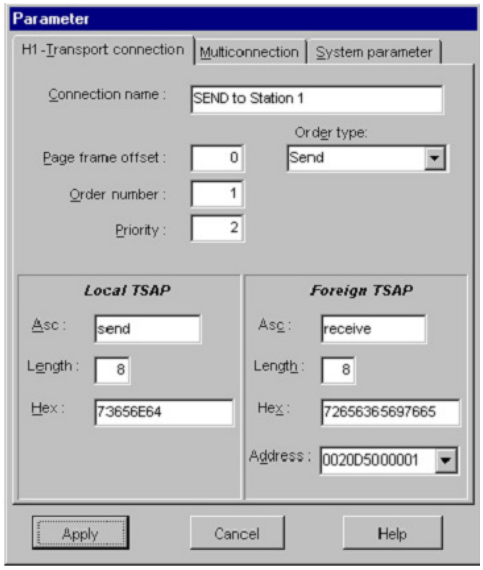
Станція 2
Отримання від Станції 1



Станція 1
Отримання від Станції 2



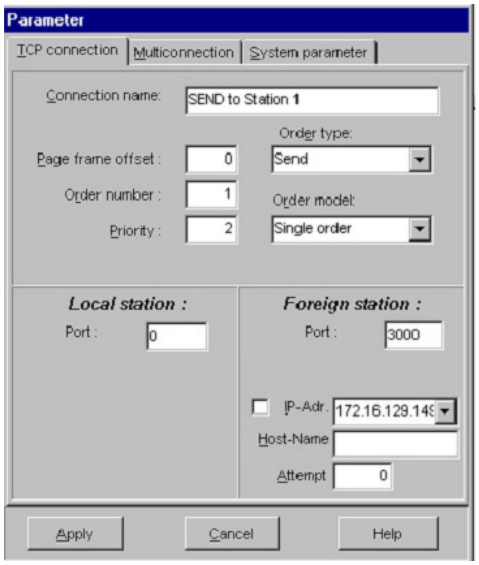
Станція 2
Відправка на адр.: 0020D5000001



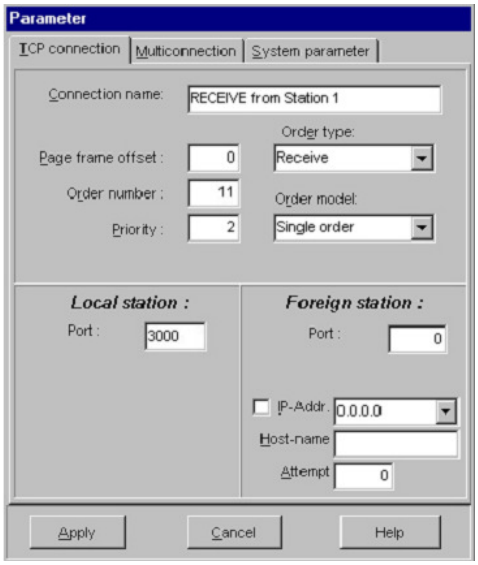
TCP/IP

Конфігурування комунікації TCP/IP виконується натисненням кнопки  і конфігуруванням відповідних вікон параметризації.

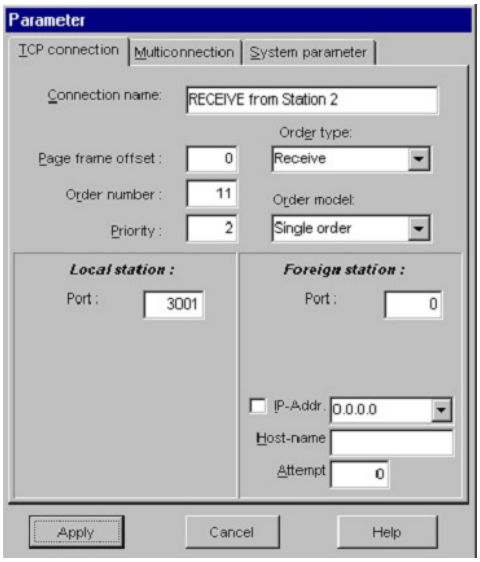
Станція 1
Відправка на IP: 172.16.129.149



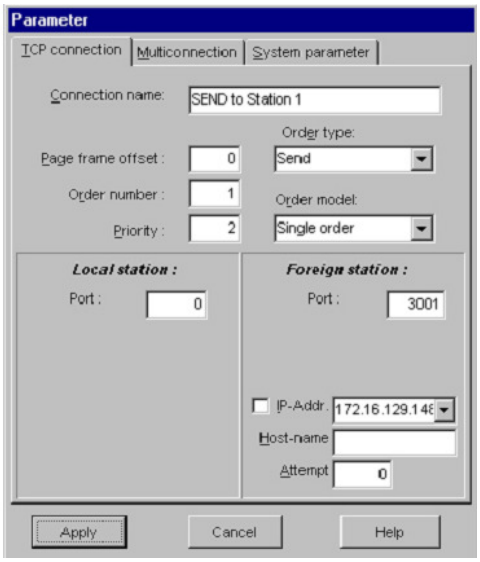
Станція 2
Отримання від Станції 1



Станція 1
Отримання від Станції 2



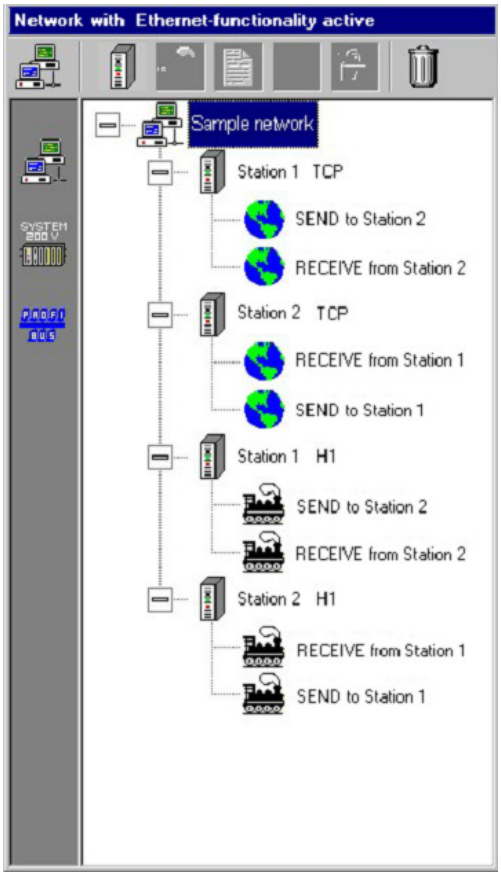
Станція 2
Відправка на IP: 172.16.129.148



Збережіть проект!

Мережеве вікно

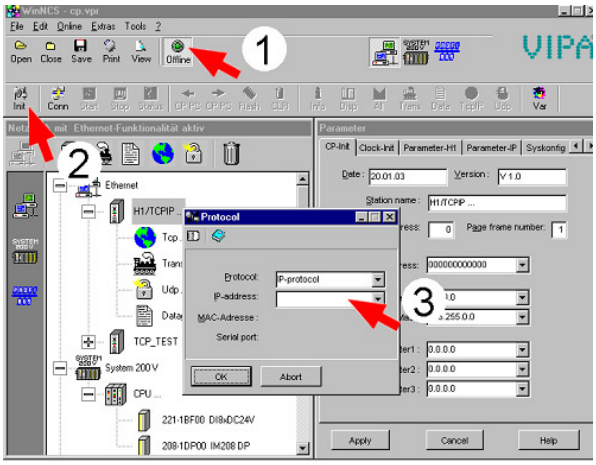
Ваше вікно відображення мережі повинне мати наступний вигляд:



Перенесення конфігураційних даних в CPU

Передача конфігурації в CPU можете бути виконана в онлайн режимі через мережу. Створіть структуру системи, як показано вище, і запустіть обидва CPU.

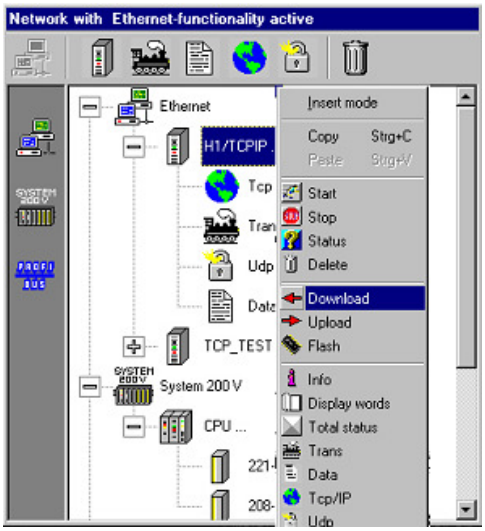
Для передачі даних Вам необхідно активувати онлайн функцію і натиснути кнопку INIT:



Тепер у вікні *Protocol* виберіть "IP-протокол" і введіть відповідну IP адресу. Натисніть [OK].

У вікні Network в WinNCS виберіть станцію на яку Ви хочете перенести дані конфігурації і активуйте "Download".

Тепер Ваш проект буде перенесено в RAM CPU.

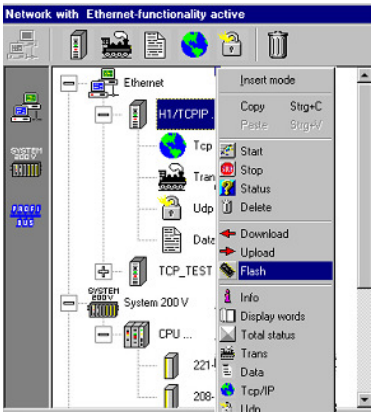


Увага!

Після передачі проекту в CPU-RAM, додатково необхідно зберегти проект в Flash-ROM. В іншому випадку при вимкненні живлення дані будуть втрачені.

У вікні Network найдіть відповідну станцію. Правою кнопкою миші викличте контекстне меню і виберіть "Flash".

Таким чином дані RAM будуть передані в Flash-ROM.



Повторіть процедуру описану вище для станції 2.

На цьому конфігурування секції CP буде закінчено. Далі докладно описано програмування секції PLC.

PLC програми для CPU

Програмування PLC в даному випадку не залежить від протоколу і тому може бути використано для ISO-транспорту і TCP/IP.

Ця програма PLC використовується в обох процесорах.

Синхронізація інтерфейсу OB100

Синхронізація інтерфейсу

Під час запуску процесора інтерфейс, який використовує CP має бути синхронізований з допомогою блоку обробки SYNCHRON.

OB100 підтверджує, що процедура синхронізації була завершена без помилок. Якщо це не так, номер помилки заноситься в MB200.

Операційний блок OB100:

```
OB100 :  "Complete Restart"
Commentary:

Network  1: Title:
Commentary:

      L     S5T#20S                 // Timer 20sec
      CLR                           // VKE 0
      SV    T     1
      SET                           // VKE 1
      SV    T     1                 // Timer Start

loop: CALL  "Synchron"              // CP synchron
       SSNR:=0                      // Interface number
       BLGR:=6                      // Block size 6 : 512 byte
       PAFE:=MB200                  // Configuration error byte

      UN    M     200.0             // no error - ready
      BEB
      U     T     1                 // Timer running
      SPB   loop                    // recall synchron
```

OB1 - Cycle
FC 1 - SEND
FC 2 - RECEIVE

Цикл OB1 контролює відправку та отримання даних. Ініціація передачі в станції 1 відбувається за допомогою блоку обробки SEND, який викликається в FC1. Партнерська станція відповідає блоком RECEIVE (FC2). За допомогою команди SEND_ALL дані будуть відправлені, і отримані партнером через RECEIVE_ALL.

Операційний блок Cycle - OB1:

```
OB1 : Cycle
Send-All, Rec-All, Applikation

Network 1: All Function
Send- und Rec-All

      CALL  "Receive_All"           // takes data at CP
       SSNR:=0                      // is also used write passiv
       PAFE:=MB196                  // Parametrization error byte MB196
       ANZW:=MD180                  // Indication word:
//                                     showes the working order number

      CALL  "Send_All"              // send data at CP
       SSNR:=0                      // is also used from Fetch passiv
       PAFE:=MB197                  // Parametrization error byte
       ANZW:=MD184                  // Indication word:
//                                     showes the working order number

Network 2: Title
SEND

      CALL  FC  1                   // Function call 1: sending


Network 3: Title
RECEIVE

      CALL  FC  2                   // Function call 2: receive
```

FC1 - SEND

```
FC1 : Send
Send function

Network 1 : Title
Time control sending

      CALL  "Control"               //control send
       SSNR:=0                      //Interface number 0
       ANR :=1                      //Order number 1
       PAFE:=MB195                  //Parametrization error MB195
       ANZW:=MD174                  //Indication word MD174

      0     M    175.1              //Job runs
      0     T      1                //Timer runs
      BEB                           //end

      L     S5T#1S                  // Timer 1 sec
      CLR                           // vke 0
      SV    T      1
      SET                           // vke 1
      SV    T      1                // Timer start

      CALL  "Send"                  // call send
       SSNR:=0                      // Interface number 0
       ANR :=1                      // Order number 1
       IND :=0                      // Indirect configuration
       QANF:=P#DB11.DBX 0.0 BYTE 16 // Data : db11, dw0, 16 byte
       PAFE:=MB196                  // Parametrization error MB 196
       ANZW:=MD174                  // Indicator word MD 174

      L     DB11.DBW    0           // Message counter
      +     1                       // DB11, DW0
      T     DB11.DBW    0           // increment message counter
```

FC2 - RECEIVE

```
FC2 : RECEIVE
RECEIVE funktion

Network 1 : Title
Cycle

      CALL  "Control"               // SFC233
       SSNR:=0                      // Interface number 0
       ANR :=11                     // Order number 11
       PAFE:=MB195                  // Parametrization error MB 195
       ANZW:=MD184                  // Indicator word MD 184

      UN    M    185.0              // if no handshake
      BEB                           // end


      CALL  "Receive"               // SFC231
       SSNR:=0                      // Interface number 0
       ANR :=11                     // Order number 11
       IND :=0                      // Indirect configuration
       ZANF:=P#DB12.DBX0.0 BYTE 16
       PAFE:=MB195                  // Parametrization error MB 195
       ANZW:=MD184                  // Indicator word MD 184

      L     MB   188                // Increment message counter
      +     1
      T     MB   188
```

Блоки даних DB11, DB12

Частота виконання команди SEND залежить від інтенсивності виклику FC1. В розглянутому прикладі цей параметр встановлено в 1000мс. Тобто виконання команди SEND ініціюється один раз в 1000мс.

При кожному виконанні команди SEND слово DW0 в блоці даних DB11 збільшується. Це відбувається в функціональному блоці FC1. Усього передається 16Byte даних. Партнер отримує інформацію і зберігає її в DB12. Разом з DB0 залишається іще 15Byte вільного місця для даних користувача.

DB11 і DB12 ідентичні по структурі:

Address	Name	Type	Startvalue	Comment
0.0		STRUCT		
+0.0	STAT0	BYTE	B#16#0	
+1.0	STAT1	BYTE	B#16#0	
+2.0	STAT2	BYTE	B#16#0	
+3.0	STAT3	BYTE	B#16#0	
+4.0	STAT4	BYTE	B#16#0	
+5.0	STAT5	BYTE	B#16#0	
+6.0	STAT6	BYTE	B#16#0	
+7.0	STAT7	BYTE	B#16#0	
+8.0	STAT8	BYTE	B#16#0	
+9.0	STAT9	BYTE	B#16#0	
+10.0	STAT10	BYTE	B#16#0	
+11.0	STAT11	BYTE	B#16#0	
+12.0	STAT12	BYTE	B#16#0	
+13.0	STAT13	BYTE	B#16#0	
+14.0	STAT14	BYTE	B#16#0	
+15.0	STAT15	BYTE	B#16#0	
+16.0	STAT16	BYTE	B#16#0	
=18.0		END_STRUC		

Передача проекта

Передача даних здійснюється через MPI. Якщо програматор не має MP-інтерфейсу, для передачі проекта Ви можете використати "Green Cable" (VIPA 950-0KB00) від VIPA.

"Green Cable" може використовуватися тільки в процесорах VIPA системи 100V, 200V, 300V і 500V!

Детальніше дивіться технічну документацію "CPU 21X" розділ "Передача проекту".

• Підключіть PG до CPU

• Через меню конфігуратора **PLC** > *Load to module* проект передається в CPU

• У відповідний слот CPU вставте Multi Media Card (MMC) і передайте програму через меню **PLC** > *Copy RAM to ROM*

• Під час процесу запису MC-LED індикатор CPU буде блимати. Успішний запис проекту підтверджується відповідним повідомленням в CPU. По завершенню процесу світлодіод тухне

• Переведіть обидва CPU в режим RUN.

Моніторинг передачі даних в Siemens SIMATIC Manager

Передбачається, що CP запрограмовані, виконано загальне скидання CPU, і перемикач RUN/STOP переведено в положення СТОП.

Завантажте представлену вище програму PLC в CPU. Запустіть її на виконання шляхом переводу перемикача RUN/STOP в положення RUN.

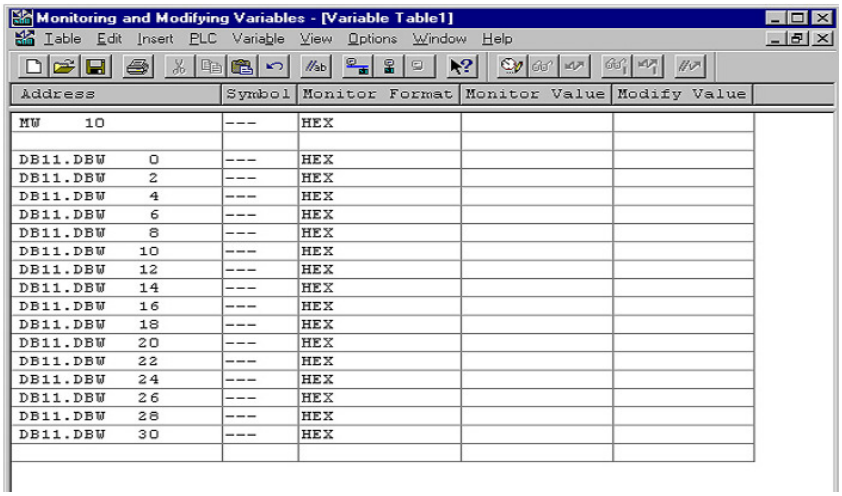
В цей момент між модулями повинен встановитися зв'язок. Про це сигналізує COMM-LED індикатор.

Запустіть Siemens SIMATIC Manager і виконайте наступні кроки для моніторингу процесу обміну даними:

• **PLC** > *Monitor/Modify Value*

• В колонці "Operand" введіть номер необхідного блоку даних і слово даних (DB11.DW0-15).

• Встановіть з'єднання і натисніть кнопку "Monitor" .



Введення даних користувача

Користувач можете ввести свої дані починаючи з DW1. Активуйте колонку *Modify value* і введіть значення, яке Ви хочете передати, наприклад, W#16#1111.

Кнопка  передає нове значення в кожному циклі, а кнопка  ініціює одноразову передачу.

Стадія запуску

Короткий огляд

При підключенні напруги живлення CPU і CP виконують процедуру BIOS (апаратна та драйверна ініціалізація і тест пам'яті).

У той час, коли CPU розпізнає встановлені на внутрішній шині модулі і завантажує програму, CP починає процедуру ініціації адміністрування.

Через приблизно 15сек. CP очікує на надходження від процесора запиту синхронізації. У зв'язку з цим комунікація з CPU блокується і поновлюється тільки після виконання синхронізації.

Загальний час завантаження CPU 21x-2BT02, в тому числі CP, складає приблизно 18сек.

Статус після завантаження CP

При кожній зміні статусу від STOP до RUN чи навпаки, CPU 21x-2BT02 виконує холодне\гаряче скидання. Оскільки CP перезавантажується всі з'єднання очищуються і повторно відновлюються.

Запити зміни статусу можуть бути згенеровані по трьом різним причинам:

• Ресинхронізація CP через SYNCHRON-HTB CPU (теплий старт)

• STOP/START-функція конфігураційного інструменту WinNCS (теплий старт)

• RESET_ALL (теплий старт).

Примітка!

Якщо в CP передано некоректну конфігурацію і він не може стартувати, вона буде стерта і встановлено значення по-замовчуванню. При цьому CP отримає початкову IP адресу.

Це дозволяє очистити CP незалежно від CPU.

Системні властивості CPU 21x-2BT02

Примітка

Системні властивості CP не слід розглядати як обмеження або прирівнювати до несправностей. Деякі функції не підтримуються або небажані, якщо система розглядається вцілому.

Загальна інформація

Час завантаження CP-секції CPU 21x-2BT02 складає приблизно 18 секунд.

Вбудований блок SYNCHRON (затримка 30сек.) виконується під час завантаження.

Примітка для ISO-транспорту (H1)

• Команди з пріоритетом 0/1 можуть передати та/або отримати максимальну кількість даних яка визначена в SYNCHRON-HTB. Команди цього пріоритету не блокуються. В результаті максимальна швидкість передачі даних складає 512Byte для блоків розміром 255 (див. також розмір блоку).

• Команди RECEIVE, які перерозподілені до широкомовної передачі даних не можуть отримати всі дані повідомлень від передавача в швидкому циклі передачі. Повідомлення, які не були отримані, відкидаються.

Примітка для TCP/IP

• Довжина за замовчуванням (-1, 0xFFFF) не допускається для ORG-формату, тобто користувач повинен визначити точну довжину даних, які будуть отримані.

• Команди з пріоритетом 1 можуть передати та/або отримати максимальну кількість даних яка визначена в SYNCHRON-HTB. В результаті максимальна швидкість передачі даних складає 512Byte для блоків розміром 255 (див. також розмір блоку).

• Команди RECEIVE, які перерозподілені до комунікаційного типу UDP не можуть отримати всі дані повідомлень від передавача в швидкому циклі передачі. Повідомлення, які не були отримані, відкидаються.

Стек протоколу TCP/IP має глобальний набір буферів, де зберігаються вхідні\вихідні буфери. Колізія в цій системі може трапитись якщо:

• Дані про приймання команд не накопичуються. Після певного періоду роботи комунікації через недостачу ресурсів пам’яті інші з'єднання припинять себе.

Після очищення буфера з’єднання буде відновлене. Також це відбудеться якщо дані були отримані за допомогою команди RECEIVE.

• Одна чи більше циклічних станцій перевантажили CP. При одночасному намаганні передати дані одній станції, CP може ініціювати припинення з’єднання.

• Станція передає два або більше повідомлень і приймач не має можливості їх прийняти. Прийом невідомого типу даних призведе до колізії в приймачі. Тим не менш, CP перешкоджає цьому. Для прийому даних додаток PLC вимагає наявності певного об’єму пам’яті. Використання шаблонної довжини не допускається. Розмір приймального блоку Prio1-RECEIVE визначається непрямим чином через наперед визначені розміри блоків (16, 32, 64, 128, 256, 512Byte).

• Для забезпечення передачі даних VIPA рекомендує використовувати підтверджувальні повідомлення.

• Майте на увазі, що **Порт 7777** використовується WinNCS для зв'язку. Контролюйте щоб він не був зайнятий іншими додатками!

Поєднання з іншими системами

ORG-формат

Організаційний формат представляє собою скорочений опис джерела даних або їх призначення в середовищі PLC. Нижче наводяться доступні формати ORG.

При виконанні READ та WRITE блок ORG не є обов'язковим.

Ідентифікатор ERW використовується для адресації блоків даних. В цьому випадку номер блока даних заноситься до цього ідентифікатора. Початкова адреса і кількість вказує на адресу області пам'яті. Вони зберігаються в форматі HIGH/LOW (Motorola формат адрес).

Опис	Тип	Діапазон
ORG ідентифікатор	BYTE	1..x
ERW ідентифікатор	BYTE	1..255
Початкова адреса	HILOWORD	0..y
Довжина	HILOWORD	1..z

Наступна таблиця містить список доступних форматів ORG:

ORG ідентифікатор 01h-04h

Область CPU	DB	MB	EB	AB
ORG ідентифікатор	01h	02h	03h	04h
Опис	Вхідні\вихідні дані від\в блок даних в головній пам'яті	Вхідні\вихідні дані від\в область меркерів	Вхідні\вихідні дані від\в образ процесу входів (PAE)	Вхідні\вихідні дані від\в образ процесу виходів (PAA)
ERW ідентифікатор (DBNR) дійсний діапазон:	DB з якого отримані вихідні дані чи куди дані призначення передаються 1...255	не має значення	не має значення	не має значення
Початкова адреса Значення дійсний діапазон:	Номер DB з якого отримуються дані чи куди вони зберігаються 0...2047	Номер MB з якого отримуються дані чи куди вони зберігаються 0...255	Номер EB з якого отримуються дані чи куди вони зберігаються 0...127	Номер AB з якого отримуються дані чи куди вони зберігаються 0...127
Довжина Значення дійсний діапазон:	Довжина вихідного\призначення блоку даних в форматі слова 1...2048	Довжина вихідного\призначення блоку даних в форматі слова 1...256	Довжина вихідного\призначення блоку даних в форматі слова 1...128	Довжина вихідного\призначення блоку даних в форматі слова 1...128

Структура заголовку PLC

Для кожної команди READ та WRITE CP генерує PLC-заголовки для повідомлень запиту та підтвердження. Зазвичай довжина цих заголовків 16Byte, і вони мають наступну структуру:

для WRITE

Повідомлення запиту

Системний ідентифікатор	="S" ="5"
Довжина заголовку	=16d
Ідент. OP-код	=01
Довжина OP-коду	=03
OP-код	**=03**
ORG блок	=03
Довжина ORG блоку	=08
ORG ідентифікатор	
ERW ідентифікатор	
Початкова адреса	*H* *L*
Довжина	*H* *L*
Буферний блок	=FFh
Довжина буферного блоку	=02
64K даних лише якщо помилка	=0

Повідомлення підтвердження

Системний ідентифікатор	="S" ="5"
Довжина заголовку	=16d
Ідент. OP-код	=01
Довжина OP-коду	=03
OP-код	**=04**
Блок підтв.	=0Fh
Довжина блоку підтв.	=03
Номер помилки	=No.
Буферний блок	=FFh
Довжина буферного блоку	=07
не використовується	

для READ

Повідомлення запиту

Системний ідентифікатор	="S" ="5"
Довжина заголовку	=16d
Ідент. OP-код	=01
Довжина OP-коду	=03
OP-код	**=05**
ORG блок	=03
Довжина ORG блоку	=08
ORG ідентифікатор	
ERW ідентифікатор	
Початкова адреса	*H* *L*
Довжина	*H* *L*
Буферний блок	=FFh
Довжина буферного блоку	=02

Повідомлення підтвердження

Системний ідентифікатор	="S" ="5"
Довжина заголовку	=16d
Ідент. OP-код	=01
Довжина OP-коду	=03
OP-код	**=06**
Блок підтв.	=0Fh
Довжина блоку підтв.	=03
Номер помилки	=No.
Буферний блок	=FFh
Довжина буферного блоку	=07
не використовується	
64K даних лише якщо помилка	=0

SEND/RECEIVE тип TRADA

TRADA контролює Прозорий Обмін Даними. Прозорий обмін даними може передавати дані додатків з різними довжинами блоків. 16Byte заголовку, які визначають довжину даних програми, ідуть попереду програми.

З TRADA Ви можете ввести шаблон довжини в програму PLC.

Якщо в RECEIVE-FB як довжину вели -1 (параметр: ZLAE), цим Ви визначили довжину змінної (довжина шаблону) для даних додатків. За допомогою довжини шаблону фактична довжина даних отримується з відповідного заголовку TRADA.

За допомогою функціональності TRADA наступний заголовок буде передувати команді SEND і він буде аналізуватися функцією RECEIVE.

SEND типу TRADA

OP-код = 07

Системний ідентифікатор	="S" ="5"	
Довжина заголовку	=16d	
Ідент. OP-код	=01	
Довжина OP-коду	=03	
OP-код	**=07**	
ORG блок	=03	
Довжина ORG блоку	=08	
ORG ідентифікатор		
*ERW ідентифікатор (***не має значення***)*		
Початкова адреса	*H* *L*	
Довжина	***H*** ***L***	→ Довжина даних користувача
Буферний блок	=FFh	
Довжина буферного блоку	=02	
64K даних якщо помилка =0		

Довжина

Довжина файлу містить кількість байтів в блоці даних.

При синхронізації з розміром блоку 6 (512Byte) довжина задається в форматі слова.

Програма для тестування TCP/IP комунікації

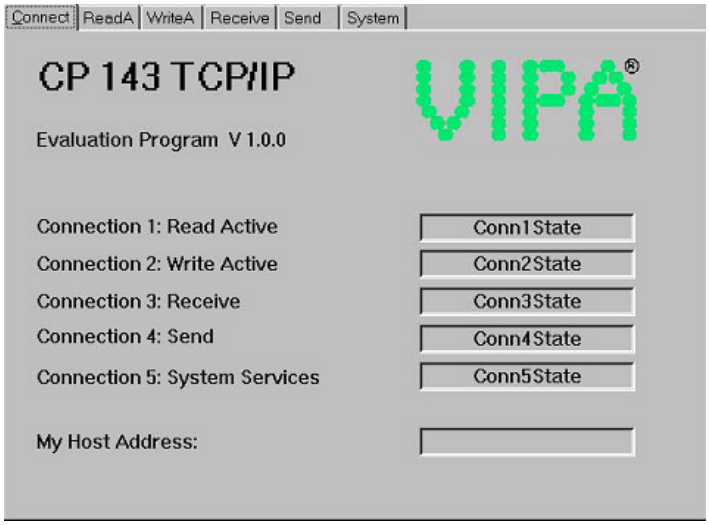
Вступ

TCPTest.exe можна знайти на ftp-сервері використовуючи посилання ftp.vipa.de/support/software. Ви можете використовувати цю тестову програму для створення простого TCP/IP з'єднання і проаналізувати його.

TCPTest не потребує додаткового встановлення, оскільки представляє собою виконавчий файл. Ця програма організовує зв'язок через Ethernet.

Наступний розділ містить короткий опис тестової програми.

Запустіть TCPTEST.EXE. Початкове вікно має наступний вигляд:



Розділи програми

Меню представляє собою набі закладок, які відкриваються натисненням лівої кнопки миші.

Connect	Вікно містить статус з'єднання та локальну IP-адресу
ReadA	Вікно конфігурування READ AKTIV з'єднань (FETCH)
WriteA	Вікно конфігурування WRITE AKTIV з'єднань
Receive	Вікно конфігурування RECEIVE з'єднань
Send	Вікно конфігурування SEND з'єднань
System	Контрольне вікно конфігурування запитів статусу і перемикання CP в RUN/STOP

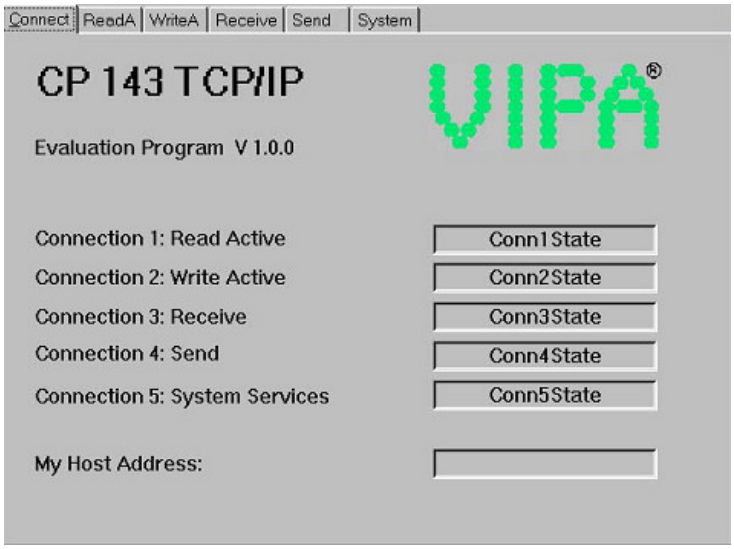
Контекстне меню (права кнопка миші)

На кожній вкладці, натиснувши правою кнопкою миші, Ви можете викликати контекстне меню.

Це меню має наступні пункти:

Save All	Збереження усіх параметрів
Save Conn1…5	Збереження відповідного з'єднання
Save Win Pos	Збереження положення вікна
Show Hints	Коли Ви розташуєте курсор в полі вводу чи на кнопці, з'явиться відповідна довідкова підказка.

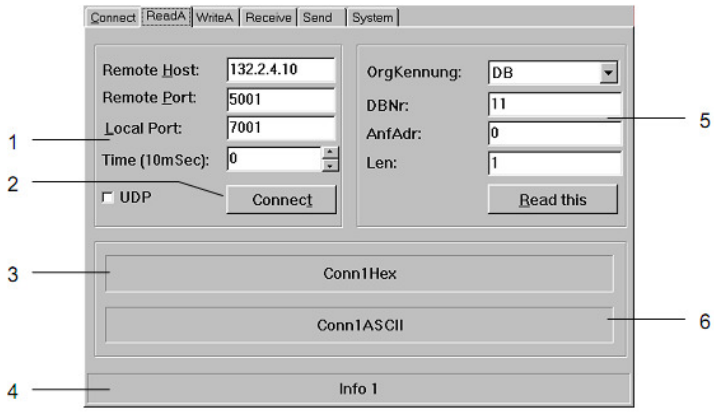
Вкладка Connect (Status)



У цьому вікні відображається стан усіх з'єднань, які можуть бути налаштовані у цій програмі. Тут видно які комунікації є стабільними, а які є нестійкими. При зміні статусу регістра, в цьому вікні відображаються відповідні зміни.

Для порівняння тут також представлена Ваша IP-адреса.

Вкладка ReadA



[1] порт
[2] встановлення з'єднання
[3] шістнадцяткове значення
[4] інформаційне вікно для відображення стану з'єднання
[5] вихідні дані
[6] отримані дані в форматі ASCII

Тут Ви можете налаштувати активне з'єднання для читання.
На додаток до даних, які потрібні для встановлення зв'язку необхідно також вказати джерело, звідки ці дані повинні бути прочитані.

Поля для вводу

Remote Host	IP-адреса станції з якої Ви бажаєте читати дані
Remote Port	Адреса Порту віддаленої станції
Local Port	Адреса Порту Вашої (локальної) станції. Для спрощення можна вказати однакову адресу порту як для віддаленої станції, так і для локальної
Time (10mSec)	Визначення інтервалу циклічності операцій читання
OrgKennung	Тип вихідного блоку
DBNr	Номер вихідного блоку
AnfAdr	Стартова адреса вихідного блоку
Len	Довжина вихідного блоку в форматі слова

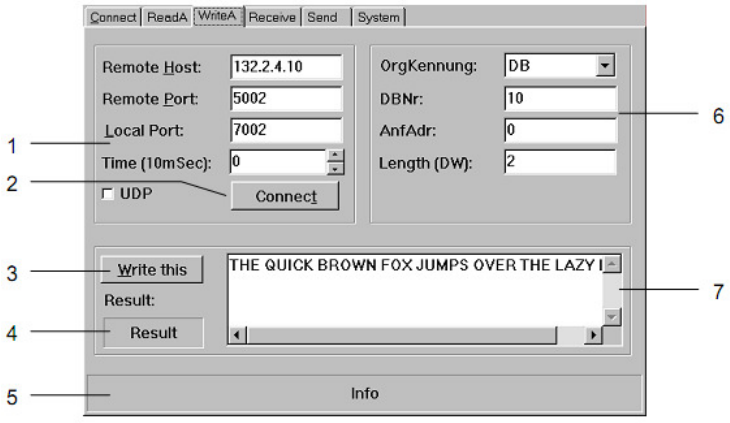
Поля активації

UDP	Це поле активує не безпечний тип зв'язку. При цьому організація віртуального з'єднання неможлива. При цьому Ви можете відображати тільки UDP-повідомлення.

Кнопки

Connect	Встановлення з'єднання та підготовка до операції читання
Read this	Читання комунікаційних даних

Вкладка WriteA



[1] порт

[2] встановлення з'єднання

[3] передача даних через з'єднання

[4] результуючий код операції запису

[5] інформаційне вікно для моніторингу статусу з'єднання

[6] вихідні дані

[7] ASCII-текст який повинний бути переданий в CP

Тут Ви можете налаштувати активне з'єднання для запису.

По аналогії з попередньою вкладкою, тут необхідно об'явити блок призначення, куди повинні бути передані дані, а також зконфігурувати параметри необхідні для встановлення комунікації.

Поля для вводу

Remote Host	IP-адреса станції на яку Ви бажаєте передавати дані
Remote Port	Адреса Порту віддаленої станції
Local Port	Адреса Порту Вашої (локальної) станції. Для спрощення можна вказати однакову адресу порту як для віддаленої станції, так і для локальної
Time (10mSec)	Визначення інтервалу циклічності операцій читання. Мінімально цей параметр може приймати значення 5
OrgKennung	Тип блоку призначення
DBNr	Номер блоку призначення
AnfAdr	Стартова адреса блоку призначення
Len	Довжина блоку призначення в форматі слова

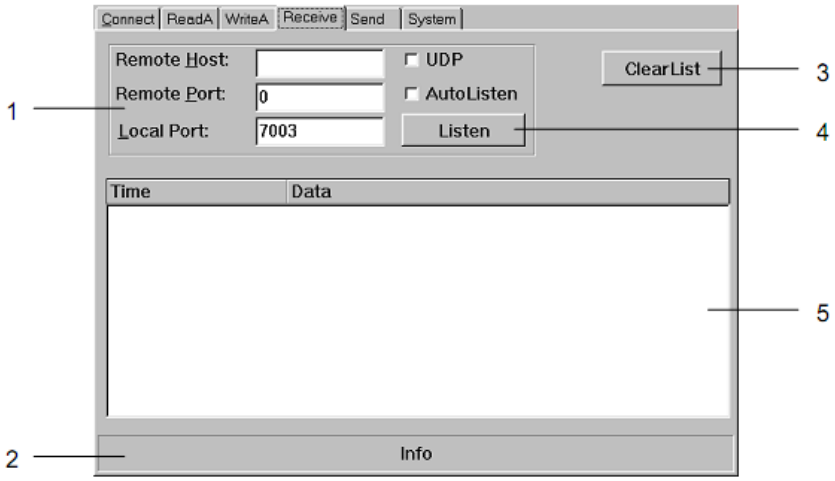
Поля активації

UDP	Це поле активує не безпечний тип зв'язку. При цьому організація віртуального з'єднання неможлива. При цьому Ви можете відображати тільки UDP-повідомлення.

Кнопки

Connect	Встановлення з'єднання та підготовка до операції запису
Write this	Дані, введені в поле ASCII, передаються в CP через встановлене з'єднання.

Вкладка Receive



[1] порт

[2] інформаційне поле для моніторингу статусу з'єднання

[3] очищення отриманого списку

[4] перегляд повідомлень

[5] перелік отриманих повідомлень

В цьому діалоговому вікні налаштовується прийом повідомлень від певного хосту.

Поля для вводу

Remote Host	IP-адреса станції де повинні зберігатися дані
Remote Port	Адреса Порту віддаленої станції
Local Port	Адреса Порту Вашої (локальної) станції. Для спрощення можна вказати однакову адресу порту як для віддаленої станції, так і для локальної

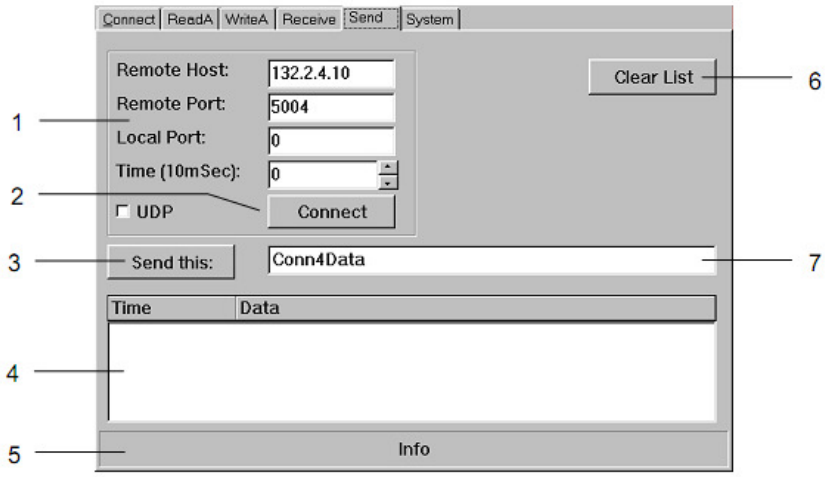
Поля активації

UDP	Це поле активує не безпечний тип зв'язку. При цьому організація віртуального з'єднання неможлива. При цьому Ви можете відображати тільки UDP-повідомлення.
AutoListen	Якщо ви виберете "AutoListen" програма перейде в режим отримання повідомлень. Кожне повідомлення, отримане від віддаленого CP відображується в списку. Перебої зв'язку виявляються і відображуються, однак, програма залишається готовою до прийому даних. Як тільки з'єднання відновлюється, повідомлення будуть знову прослуховуватись.

Кнопки

Listen	Будь-які отримані повідомлення заносяться в список. Цей процес припиниться, якщо натиснути кнопку "STOP" або з'єднання перерветься. Ви також можете зупинити заповнення списку задавши нові параметри з'єднання.
ClearList	Очищення списку отриманих повідомлень. Нові записи будуть заноситись з початку.

Вкладка Send



[1] порт

[2] встановлення з'єднання

[3] передача даних через з'єднання

[4] список переданих повідомлень

[5] інформаційне поле статусу з'єднання

[6] очищення списку повідомлень

[7] ASCII-текст який повинен бути переданий в CP

Ви можете використовувати це діалогове вікно для відправки повідомлень на певний хост.

Поля для вводу

Remote Host	IP-адреса станції де повинні зберігатися дані
Remote Port	Адреса Порту віддаленої станції
Local Port	Адреса Порту Вашої (локальної) станції. Для спрощення можна вказати однакову адресу порту як для віддаленої станції, так і для локальної
Time (10mSec)	Визначення інтервалу циклічності операцій читання. Мінімально цей параметр може приймати значення 5

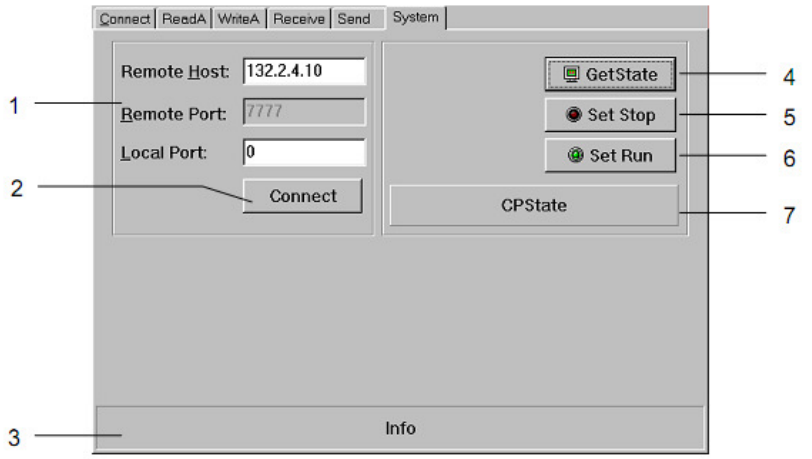
Поля активації

UDP	Це поле активує не безпечний тип зв'язку. При цьому організація віртуального з'єднання неможлива. При цьому Ви можете відображати тільки UDP-повідомлення.

Кнопки

Connect	Встановлення з'єднання та підготовка до операції запису
Send this	Дані, введені в поле ASCII, передаються в CP через встановлене з'єднання.

System Send



[1] порт
[2] встановлення з'єднання
[3] інформаційне поле статусу з'єднання
[4] статусні запити CP
[5] CP в STOP
[6] CP в RUN
[7] моніторинг статусу через GetState

Це діалогове вікно дає інформацію про визначений хост-CP.

Поля для вводу

Remote Host	IP-адреса станції де повинні зберігатися дані
Remote Port	Комунікаційна адреса віддаленої станції
Local Port	Адреса Порту Вашої станції. Для спрощення можна вказати однакову адресу порту як для віддаленої станції, так і для локальної

Кнопки

Connect	Встановлення з'єднання та підготовка до операції запису
GetState	Через з'єднання, встановлене за допомогою *Connect*, передається і моніториться статус CP у вікні статусу Контролюються наступні параметри: - Hardware-Stop (Перемикач Run/Stop на CP в положенні Stop) CP не повинен бути відчуженим програмою - Hardware-Run (Перемикач Run/Stop на CP в положенні Run) CP повинен бути відчуженим програмою - Software-Stop (Перемикач Run/Stop на CP в положенні Run) CP повинен бути переведений в Stop за допомогою *SetStop* - Software-Run (Перемикач Run/Stop на CP в положенні Run) CP повинен бути переведений в Start за допомогою *SetRun*.
SetStop	CP переведено в Stop. Ця функція доступна лише, якщо CP-перемикач знаходиться в положенні Run
SetRun	CP переведено в Run. Ця функція доступна лише, якщо CP-перемикач знаходиться в положенні Run.