Розділ 9 CPU 21xCAN

Короткий огляд

Цей розділ описує застосування CPU 21xCAN в мережах CANopen. Тут Ви знайдете усю необхідну інформацію для використання інтегрованого CAN-майстра.

Зміст

Тема	Сторінка
Розділ 9 CPU 21xCAN	**9-1**
Базові положення CAN-шини	9-2
Розробка проекту для CPU 21xCAN	9-4
Режими	9-13
Образ процесу CPU 21xCAN	9-14
Повідомлення CANopen	9-16
Об’єктна директорія	9-21

Базові положення CAN-шини

Загальна інформація

CANopen (**C**ontrol **A**rea **N**etwork) - міжнародний стандарт для відкритих польових систем, який широко використовується в будівництві, виробництві та промисловій автоматизації. Спочатку CAN був розроблений для застосування в автомобілебудуванні.

Завдяки наявності широких засобів виявлення помилок, мережа CAN є найбезпечнішою магістральною системою. Імовірність залишкової похибки становить менш ніж $4.7x10^{-11}$. “Біти” повідомлення маркуються мітками і автоматично ретранслюються.

На відміну від Profibus і Interbus, мнемоніка CAN визначається за CAL-level-7-протоколом (CAL = **C**AN **a**pplication **l**ayer), який регламентує створення 7 рівнів користувацьких профілів мережі CAN. CANopen - це стандартний користувацький профіль CIA (**C**AN **i**n **A**utomation).

CANopen

CANopen - це користувацький профіль для індустріальних систем реального часу, який в даний момент підтримується великою кількістю виробників. CANopen був створений під назвою DS-301 організацією CAN in Automation (CIA). Специфікації комунікацій DS-301 визначають стандарти для CAN пристроїв. Взаємозамінність обладнання різних виробників декларується в специфікації DS-401. Тут визначаються стандарти для входів\виходів дискретних і аналогових модулів.

CANopen включає комунікаційний профіль, що визначає об'єкти, які повинні використовуватись для передачі даних, та профілі пристроїв, які конкретизують вид даних що передаються до об'єктів.

Комунікаційний профіль CANopen грунтується на об'єктній директорії, подібній до профілів Profibus. Комунікаційний стандарт DS-301 визначає два стандартних об'єкта і ряд спеціальних об'єктів:

- Процесні об'єкти даних (PDO)

  PDO використовується для передачі даних в режимі реального часу

- Сервісні об'єкти даних (SDO)

  SDO забезпечує доступ до об'єктної директорії для операцій читання і запису

Середовище передачі даних

CAN базується на лінійній шинній топології. Для побудови мережі Ви можете використовувати маршрутизатори. Кількість пристроїв в мережі обмежується продуктивністю шинних драйверів відповідних модулів. Максимальна відстань, що покривається мережею, залежить від об'єму даних. Так, якщо передається 1Mbit/s, мережа обмежується 40м, а при 80kbit/s мережа обмежується 1000м. В CAN в якості середовища передачі даних використовується екранований трижильний кабель (при необхідності - п'яти). CAN використовує диференціальні напруги. Тому вона менш чутлива до зовнішніх перешкод, ніж мережа, що використовує напругу або струм. Мережа конфігурується як послідовна шина, на кінцях якої розміщуються термінальні резистори 120Ω.

Шинний розподілювач VIPA CAN має 9-піновий роз'єм, який використовується для підключення модуля як слейв в мережу CAN.

Всі мережеві пристрої використовують однакову швидкість передачі даних.

Підключення нових станцій або їх відключення виконується без втручання в систему. Пошкоджені або нові станції розпізнаються в мережі автоматично.

Метод доступу до шини

Шинні методи доступу зазвичай поділяються на керовані (детерміновані) і некеровані (випадкові).

CAN використовує метод множинного доступу з опитуванням несучої (CSMA), тобто всі станції мають однакові права доступу до шини, поки шина не зайнята (випадковий доступ).

Передача даних орієнтована на повідомлення, а не на станції. Кожне повідомлення містить унікальний ідентифікатор, який також визначає і пріоритет повідомлення. Одночасно доступ до шини може мати тільки одна станція.

Доступ до CAN-шини здійснюється за методом уникнення колізій, використовуючи алгоритм побітового розподілу. Уникнення колізій означає, що кінцева станція, яка в даному випадку має пріоритет, не матиме необхідності повторювати своє повідомлення. Якщо до шини одночасно звертається декілька станцій, то доступ надається в порядку пріоритетів. Якщо шина зайнята, будь яка станція, що збирається передавати інформацію, повинна перервати цю процедуру.

Розробка проекту для CPU 21xCAN

Короткий огляд

Розробка проекту для CANopen-майстра відбувається в WinCoCT (**W**indows **C**AN**o**pen **C**onfiguration **T**ool) від VIPA. Проект WinCoCT експортується як WLD-файл і після цього може бути імпортований в апаратний конфігуратор Siemens.

Створіть віртуальну PROFIBUS-систему "VIPA_CPU21x" і розмістіть CPU21xCAN (VIPA 21x-2CM02) в слот 0.

Вступ

Для проектування CAN-майстер системи на базі модулів System 200V, в каталог обладнання інструменту проектування необхідно включити відповідний GSD-файл.

Послідовність розробки проекту:

• Запустіть WinCoCT і створіть мережу CANopen.

• Кнопкою  створіть майстер-групу і через  вставте CANopen-майстра.

• В регістрі "CANopen Manager" активуйте функцію NMT-майстер і підтвердіть вибір.

• В параметрах "Set PLC Parameters" задайте діапазон адрес CPU і діагностичні дані

• Кнопкою  створіть слейв-групу і через  вставте CANopen слейва.

• Через меню "Modules" додайте в слейв-систему необхідні модулі і при необхідності запараметризуйте їх.

• Через меню "Connections" задайте необхідні з'єднання і підтвердіть зроблені дії в процесному образі майстра.

• Збережіть проект і експортуйте його в форматі WLD-файлу.

• Включіть vipa_21x.gsd в конфігуратор обладнання Siemens.

• Перейдіть в Siemens SIMATIC Manager і скопіюйте блок даних з WLD-файлу CAN в директорію блоку.

• Використовуючи Siemens-CPU в конфігураторі обладнання створіть майстер систему PROFIBUS-DP: CPU 315-2DP (6ES7 315-2AF03-0AB0V1.2)

• DP майстер отримує адресу > 1.

• Додайте слейв-систему "VIPA_CPU21x" з каталогу устаткування до існуючої майстер-системи.

• Слейв-система завжди повинна мати адресу 1.

• Встановіть усі необхідні модулі System 200V в послідовності підключення, починаючи з CPU 21xCAN у 1-му слоті.

• Збережіть і перенесіть PLC-проект разом з WLD-файлом через MPI в CPU.

Далі детально роз'яснюються ці кроки.

Передумови для створення проекту

Конфігуратор обладнання являється частиною Siemens SIMATIC Manager. При створенні проекту усі необхідні модулі можна знайти в каталозі устаткування.

Для використання System 200V в каталог обладнання необхідно включити GSD-файл vipa_21x.gsd від VIPA.



Примітка!

Для створення проектів необхідні грунтовні знання Siemens SIMATIC Manager і конфігуратора обладнання!

Вставка GSD-файлу

• Скопіюйте GSD-файл VIPA_21x.gsd в директорію \siemens\step7\s7data\gsd

• Запустіть конфігуратор обладнання Siemens.

• Закрийте всі проекти.

• Виберіть **Options** > *Install new GSD-file*.

• Виберіть **VIPA_21x.GSD**.

Тепер модулі System 200V від VIPA доступні в каталозі обладнання та можуть бути використані при проектуванні.

Примітка!

Для сумісності з Siemens SIMATIC Manager, System 200V CPU від VIPA повинні бути зконфігуровані як

CPU 315-2DP (6ES7 315-2AF03-0AB0 V1.2)!

Для прямої адресації до модулів, Ви повинні включити їх в конфігуратор обладнання від Siemens в формі віртуальної PROFIBUS системи. Для цього інтегруйте в систему відповідний GSD-файл від VIPA.

Зконфігуруйте CAN майстра в віртуальній PROFIBUS системі шляхом розміщення CPU 21xCAN у 1-му слоті.

Безпосереднє створення проекту для CANopen здійснюється за допомогою інструменту конфігурування WinCoCT. Після закінчення розробки проекту Ви можете експортувати його в форматі wld-файлу і передати як DB в PLC-програму.

WinCoCT

WinCoCT (**W**indows **C**AN**o**pen **C**onfiguration **T**ool) представляє собою інструмент конфігурування від VIPA, який розроблено для зручного створення проектів для мереж CANopen.

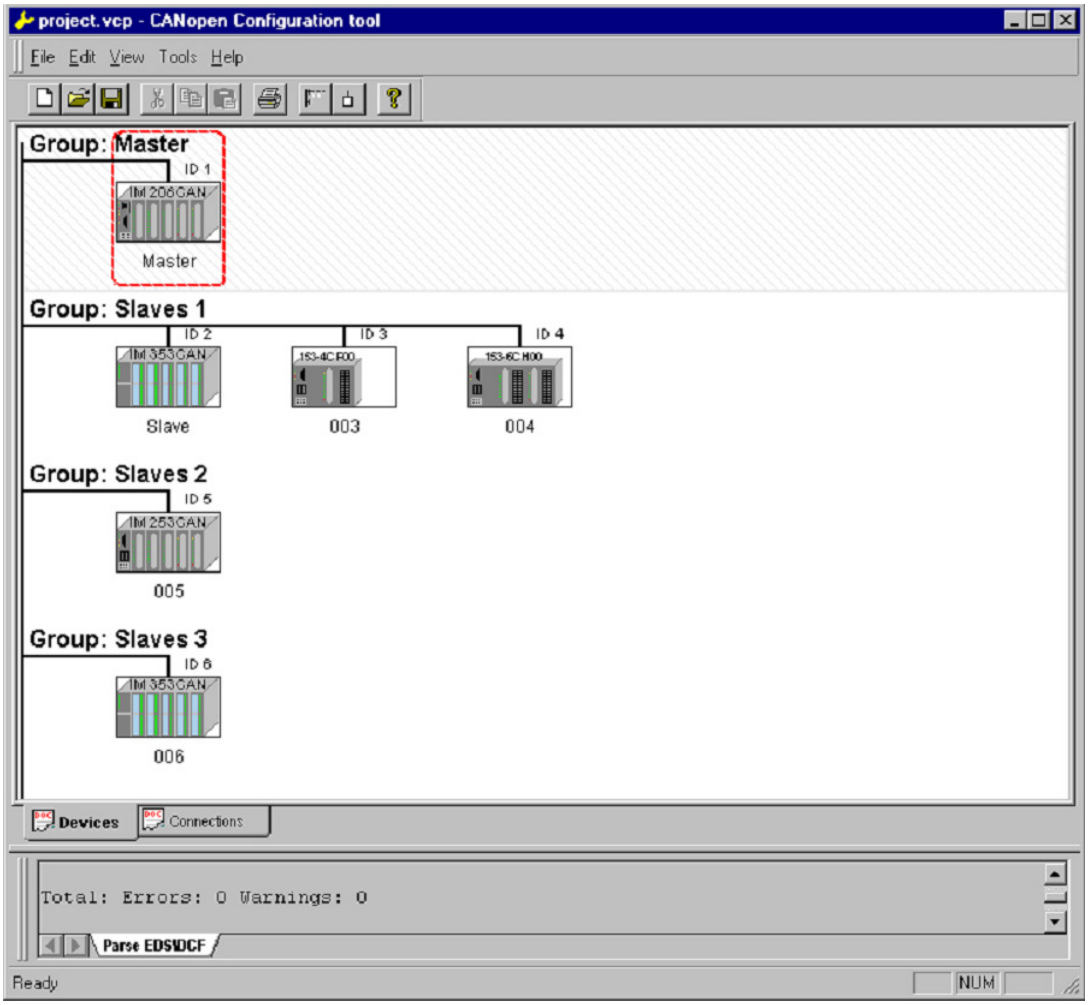
WinCoCT контролює мережеву топологію CANopen в зручному для користувача графічному виді. Тут створюються, параметризуються і групуються польові пристрої і контролери та організуються взаємні зв'язки.

Вибір пристроїв відбувається через список, який може бути розширено згідно з поставленими завданнями за допомогою EDS-файлу (**E**lectronic **D**ata **S**heet).

При натисненні правою кнопкою миші на пристрої відкриється контекстне меню, яке складається частково із статичних і частково з динамічних компонентів.

Для налаштування обміну даними процесу, система представляється в матричному вигляді: входи позначаються як рядки, а виходи – як стовпчики. На перетині утворюються точки створення потрібного з'єднання.

Опис телеграми і оптимізація комунікації виконується через WinCoCT.



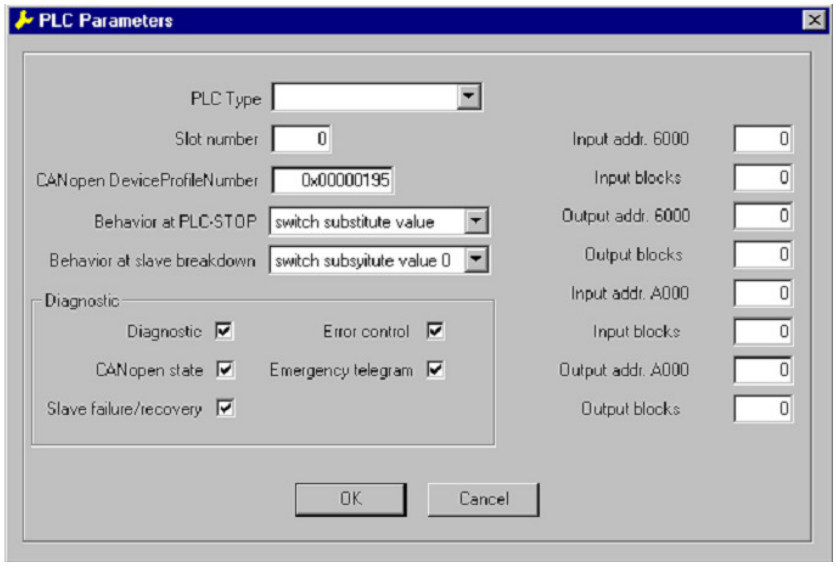
Установка параметрів проекту

Через меню **Tools** > *Project options* Ви можете задати необхідні параметри для CAN. Це такі параметри, як швидкість передачі даних, вибір майстра і т.д.

Більш детальна інформація знаходиться в технічній документації на WinCoCT.

Параметризація CAN-майстра

WinCoCT дозволяє задати параметри для CAN-майстра. Для цього необхідно з контекстного меню пристрою викликати діалогове вікно Set PLC-Parameters:



Тип PLC

Зарезервовано для подальшого розширення

Номер слоту

Номер встановлення на шині:

0: Для адресації до CAN-майстра, інтегрованого в CPU

1 ... 32: Для адресації до CAN-майстра на стандартній шині

CANopen DeviceProfileNumber

Має значення 0x195

Поведінка при PLC-STOP

Тут Ви можете визначити реакцію вихідних каналів, якщо CPU переходить в STOP. Доступні наступні значення:

Switch substitute value 0: Виходи встановлюються в 0.

Keep last value: Зберігається останній стан виходів.

Поведінка при виходу з ладу слейва

Тут Ви можете визначити реакцію вхідних каналів, якщо CPU виходить з ладу.

Switch substitute value 0: Входи встановлюються в 0.

Keep last value: Зберігається останній стан входів.

Діагностика

Ця область дозволяє визначити діагностичну реакцію CAN майстра.

Diagnostic: Активізація функції діагностування

CANopen state: При активації CAN майстер відправляє свій стан ("підготовка" чи "робота") в CPU. Цей стан можна зчитати через SFC 13.

Slave failure/recovery: При активації, у випадку виходу з ладу слейва чи перезавантаженні CPU, викликається OB 86.

Error control: Якщо вибрана ця опція, NMT-майстер відправляє всі підконтрольні помилки в якості діагностики до CPU. Список помилок можна викликати через OB 82.

Emergency Telegram: При активації NMT-майстер відправляє всі аварійні телеграми в якості діагностики до CPU. Список помилок можна викликати через OB 82.

Діапазон адрес CPU

Наступні поля дозволяють задавати діапазони адрес входів і виходів у CPU для CANopen-майстра. Кожен блок складається з 4Byte.

Вхідна адреса 6000, Вхідні блоки

PI - базові адреси в CPU, які займають 0x6000 вхідних адрес CAN. Максимально може бути задіяно 16 (64Byte) вхідних блоків.

Вихідна адреса 6000, Вихідні блоки

PO - базові адреси в CPU, які займають 0x6000 вихідних адрес CAN. Максимально може бути задіяно 16 (64Byte) вихідних блоків

Вхідна адреса A000*, Вхідні блоки*

PI - базові адреси в CPU, які займають 0xA000 вхідних адрес мережевих змінних CAN. Максимально може бути задіяно 80 (320Byte) вхідних блоків.

Вихідна адреса A000*, Вихідні блоки*

PO - базові адреси в CPU, які займають 0xA000 вихідних адрес мережевих змінних CAN. Максимально може бути задіяно 80 (320Byte) вихідних блоків.

Активація CANopen слейва в CANopen Manager

Для того щоб майстер мав доступ до CANopen слейва, Ви повинні за допомогою WinCoCT зареєструвати його у відповідного майстра. Натисніть правою кнопкою миші на CAN-майстрі і виберіть пункт "Device access" і перейдіть в реєстр "CANopen Manager".

Кнопкою [Change] реєструється окремо вибрана станція, а кнопкою [Global] реєструються усі станції, підключені до майстра і задається поведінка системи під час появи помилки.

Після завершення конфігурування системи збережіть задані налаштування натиснувши кнопку [Apply to slaves].

Кроки по розробці проекту

Далі розглянуто послідовність дій по розробці проекту на конкретному прикладі.

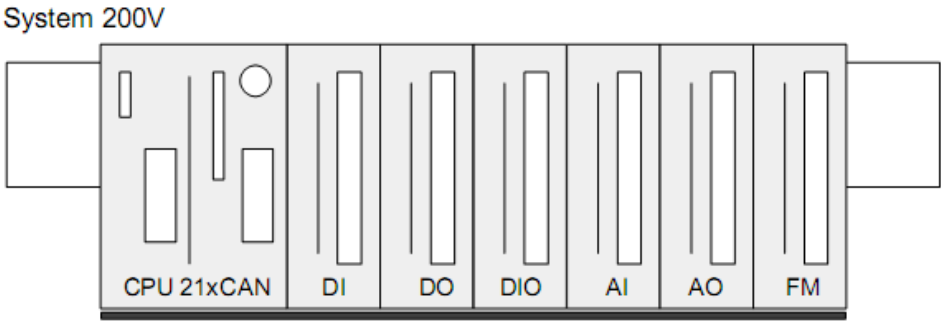
Процес розробки розділено на три частини:

• Створення проекту CAN-майстра в WinCoCT та його експорт у вигляді WLD-файлу

• Імпортування майстер-проекту в проект CPU

• Конфігурування CPU 21xCAN та модулів System 200V.

Апаратна структура

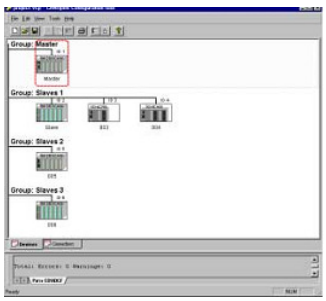


Передумови

Для успішного проектування системи CANopen, в WinCoCT необхідно імпортувати актуальний EDS-файл.

Для конфігурування модулів System 200V в апаратний каталог необхідно імпортувати GSD-файл VIPA_21x.gsd від VIPA.

Робота в WinCoCT



• Скопіюйте необхідні EDS-файли в EDS-каталог і запустіть WinCoCT.

• Кнопкою  створіть майстер-групу і через  вставте CANopen майстра (VIPA_21x_2CM02.eds).

• Кнопкою  створіть слейв-групу і через  вставте CANopen слейва.

• Через контекстне меню слейва "Modules" додайте в слейв-систему необхідні модулі.

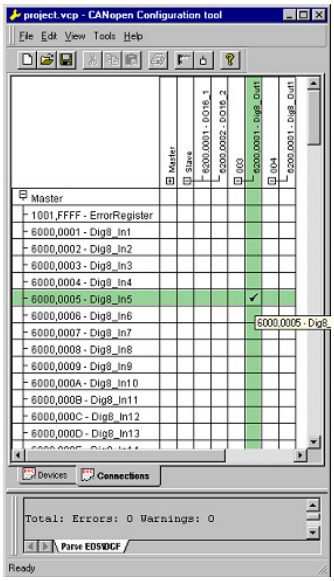
• Правою кнопкою миші викличте діалогове вікно "Device Access".

• В регістрі "CANopen Manager" активізуйте опцію Device is NMT Master та зареєструйте слейви в майстрі. За допомогою кнопки [Apply to slaves] збережіть зроблені налаштування в проекті!



• Натисніть правою кнопкою миші на CAN-майстрі і відкрийте діалогове вікно "Set PLC Parameters". Тут Ви можете налаштувати поведінку діагностики та діапазони адрес які майстер займає в CPU.

Для CPU 21xCAN в поле "Slot number" введіть номер слоту 0. При експорті WinCoCT створить DB 2000.



• В головному вікні внесіть зміни в реєстрі "Connections". Тут у вигляді матриці представлені наступні дані: входи в якості колонок і виходи в якості рядків.

Для спостереження за даними процесу за допомогою символу "+" необхідно розгорнути вузол відповідного пристрою.

• Ви можете визначити з'єднання тільки якщо рядок і стовбець, що перетинаються змінюють свій колір на зелений. Виберіть клітинку на перетині виділених полів. → Після цього вона позначиться галочкою як показано на малюнку. Ви можете контролювати з'єднання за рахунок внесення змін в "Devices". Контроль образу процесу майстра здійснюється через "Device Access".

• Збережіть проект.

• Через меню **File** > *Export* Ви можете експортувати проект CANopen в форматі WLD-файлу. Цьому файлу автоматично присвоїться ім'я, яке складається з назви проекту + адреси вузла + IDMaster/Slave.

На цьому розробку проекту CANopen в WinCoCT закінчено.

Імпорт в програму PLC

• Запустіть Siemens SIMATIC Manager з PLC-проектом для CPU 21xCAN.

• Через меню **File** > *Memory Card File* > *Open* відкрийте WLD-файл.

• Скопіюйте DB 2000 в каталог блоків даних.

Як тільки Ви передасте цей блок в CPU, він буде розпізнаний і відповідні параметри будуть передані до CAN майстра.

Це можливе тільки якщо Ваш CAN-майстер CPU зконфігуровано в апаратній конфігурації як віртуальна PROFIBUS системи. Ці дії описані нижче.

Апаратне конфігурування CPU 21xCAN і модулів System 200V

Послідовність дій по апаратному конфігуруванню System 200V:

• Запустіть конфігуратор обладнання фірми Siemens і створіть новий проект. З каталогу обладнання виберіть профільну рейку.

• Додайте CPU 315-2DP (6ES7 315-2AF03-0AB0 V1.2). Для цього створіть нову підмережу PROFIBUS.

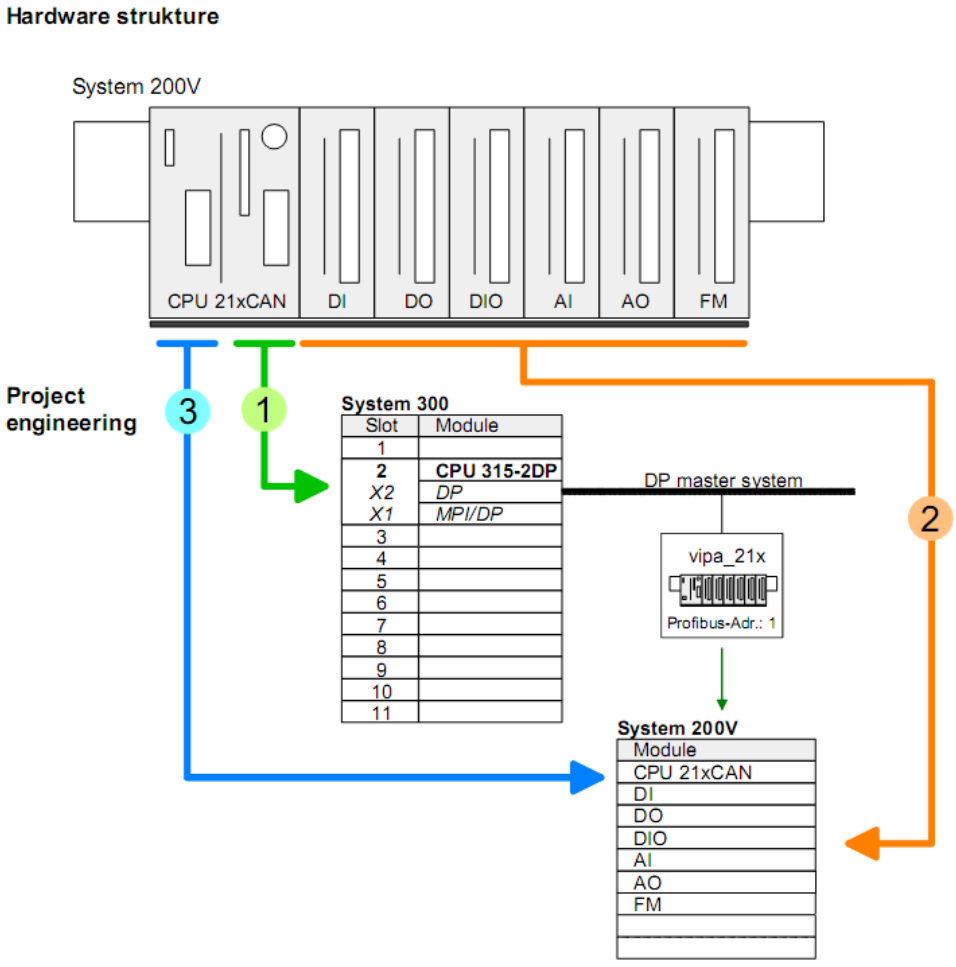
• В підмережу додайте System "VIPA_CPU21x". Ця позиція знаходиться в каталозі устаткування *PROFIBUS DP > Additional field devices > IO > VIPA_System_200V*. Цьому модулю призначте PROFIBUS-адресу 1.

• Помістіть CPU 21xCAN на 1-й слот.

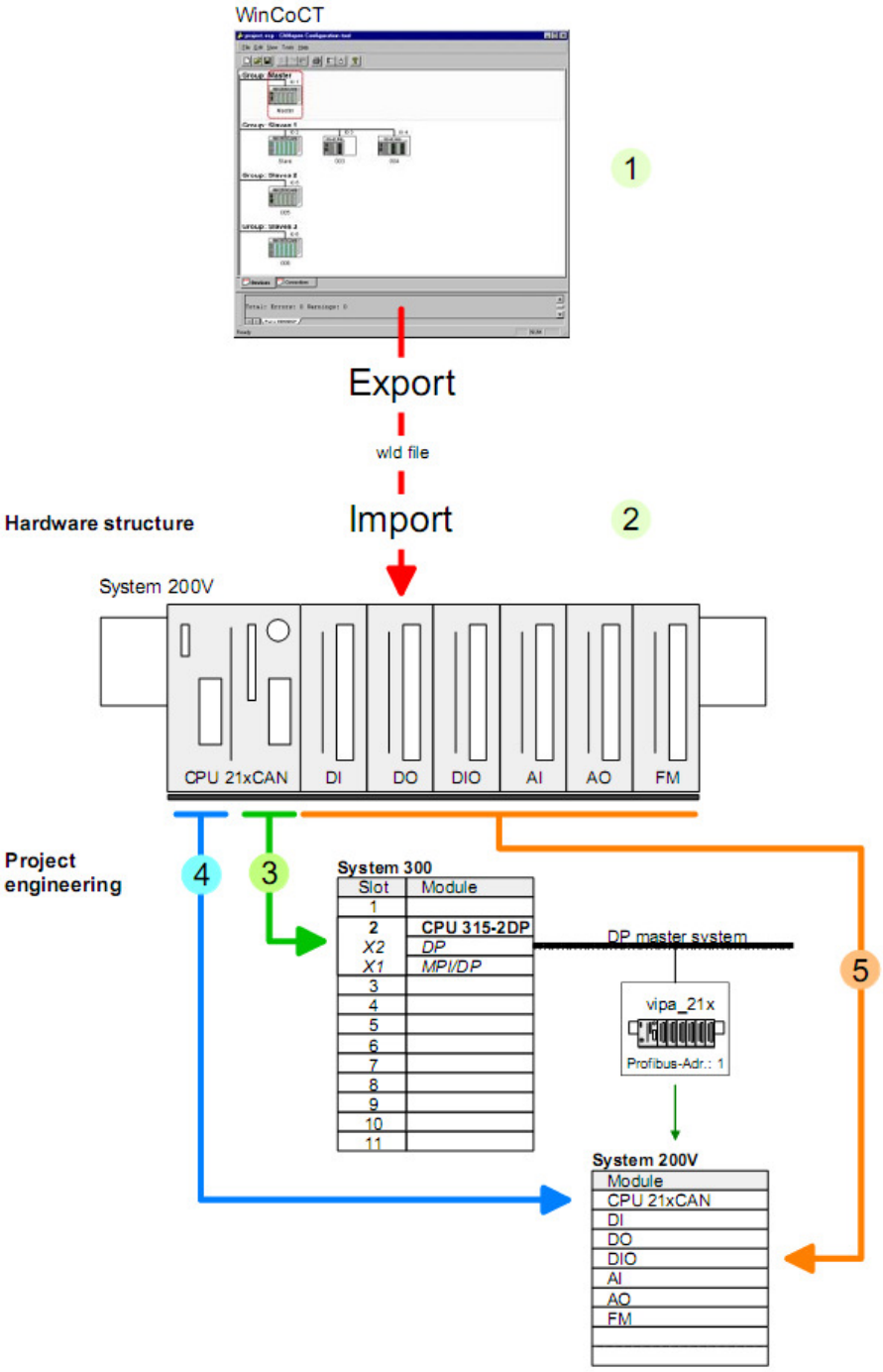
• Додайте усі необхідні модулі System 200V в послідовності їх підключення.

• При необхідності запараметризуйте CPU і модулі. Вікно параметризації відкриється при подвійному натисненні мишкою на відповідному модулі.

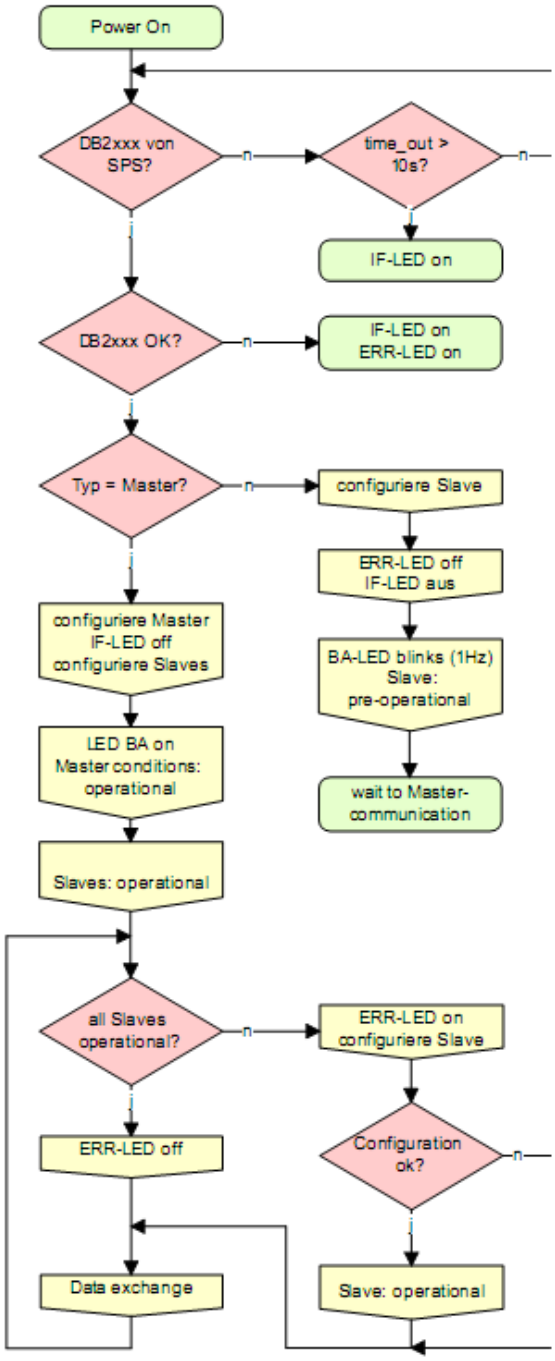
• Збережіть проект.



Заключення На наступному малюнку зображені заключні кроки по розробці проекту:



Режими



STOP → RUN (автоматично)

Після ввімкнення живлення і при достовірних даних проекту в CPU, майстер автоматично перемикається в режим RUN. Майстер не має перемикача в операційний режим.

Після ввімкнення живлення проектні дані автоматично пересилаються від CPU до CAN-майстра. При цьому встановлюється зв'язок з CAN-слейвами.

При активній комунікації та дійсними мережевими параметрами, CPU-майстер автоматично перемикається в робочий режим і вмикаються світлодіоди RUN і BA.

При некоректних параметрах, CPU-майстер переходить в режим STOP і через індикатор IF-LED діагностує помилку параметризації.

RUN

В режимі RUN горять LED-індикатори RUN і BA. Тепер контролер готовий до обміну даними.

При появі помилки, як наприклад, виходу з ладу слейва, на CPU-майстрі загорається індикатор ERR і до CPU відправляється відповідне тривожне повідомлення.

Образ процесу CPU 21xCAN

Образ процесу складається з наступних частин:

• Образ процесу вхідних даних (PI) для RPDO

• Образ процесу вихідних даних (PO) для TPDO

Кожна частина складається з 64Byte "цифрових даних" - і 320Byte "мережевих змінних".

Вхідні дані

Для вхідних даних доступні наступні об'єкти:

• 8 Bit цифровий вхід (об'єкт 0x6000)

• 16 Bit цифровий вхід (об'єкт 0x6100)

• 32 Bit цифровий вхід (об'єкт 0x6120)

• 8 Bit вхідні мережеві змінні (об'єкт 0xA040)

• 16 Bit вхідні мережеві змінні (об'єкт 0xA100)

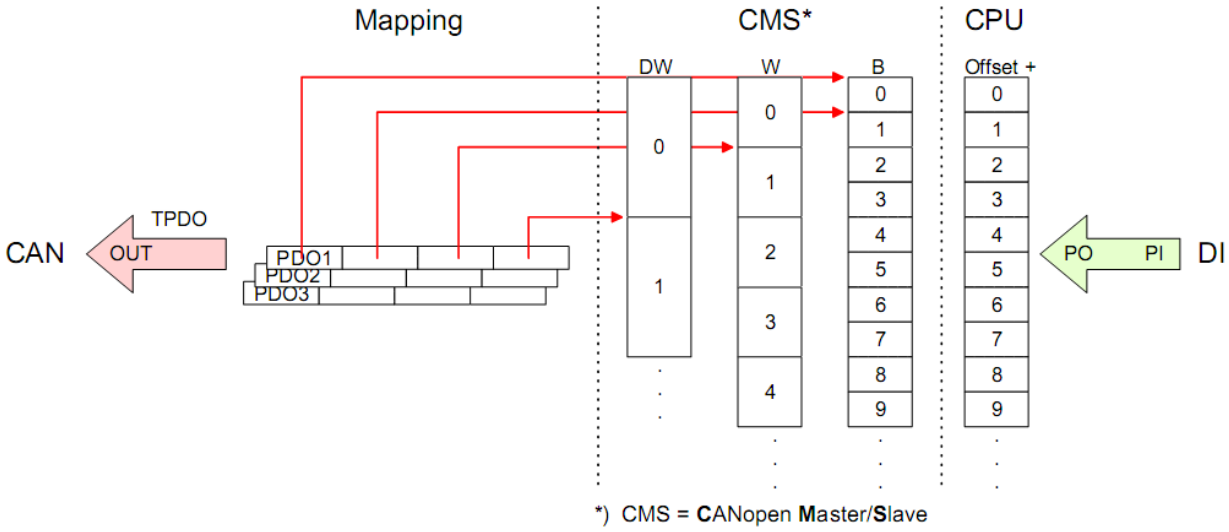
• 32 Bit вхідні мережеві змінні (об'єкт 0xA200)

• 64 Bit вхідні мережеві змінні (об'єкт 0xA440)

Як зображено на малюнку, об'єкти цифрових входів використовують суміжну область пам'яті процесора.

Наприклад, доступу до Індексу 0x6000 з Підіндексом 2 відповідає доступ до Індексу 0x6100 з Підіндексом 1. Обидва об'єкти займають в CPU ту ж комірку пам'яті.

Будь ласка, прийміть до уваги, що вхідні мережеві змінні також використовують суміжні області пам'яті.



Вихідні дані

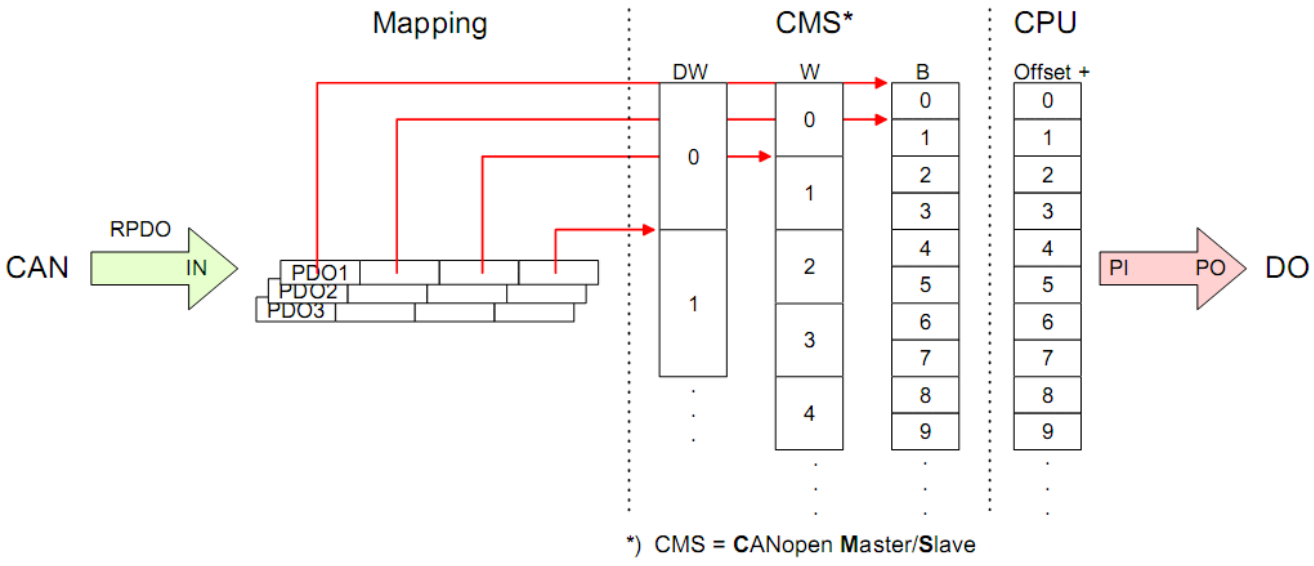
Для вихідних даних, призначення об'єктів здійснюється аналогічним чином:

- 8 Bit цифровий вихід (об'єкт 0x6200)
- 16 Bit цифровий вихід (об'єкт 0x6300)
- 32 Bit цифровий вихід (об'єкт 0x6320)
- 8 Bit вихідні мережеві змінні (об'єкт 0xA0400)
- 16 Bit вихідні мережеві змінні (об'єкт 0xA580)
- 32 Bit вихідні мережеві змінні (об'єкт 0xA680)
- 64 Bit вихідні мережеві змінні (об'єкт 0x8C0)

Як зображено на малюнку, об'єкти цифрових виходів використовують суміжну область пам'яті процесора.

Наприклад, доступу до Індексу 0x6200 з Підіндексом 2 відповідає доступ до Індексу 0x6300 з Підіндексом 1. Обидва об'єкти займають в CPU спільну комірку пам'яті.

Будь ласка, прийміть до уваги, що вхідні мережеві змінні також використовують спільні області пам'яті.



Повідомлення CANopen

Ідентифікація

Всі повідомлення CANopen, відповідно до CIA DS-301 мають наступну структуру:

Ідентифікатор

Byte	Bit 7 ... Bit 0
1	Bit 3 ... Bit 0: старші 4 біта ID модуля Bit 7 ... Bit 4: функціональний код CANopen
2	Bit 3 ... Bit 0: код довжини даних (DLC) Bit 4: RTR-Bit: 0: дані відсутні (запит коду) 1: дані присутні Bit 7 ... Bit 5: молодші 3 біта ID модуля

Дані

Дані

Byte	Bit 7 ... Bit 0
3…10	Дані

Додатковий розподіл 2Byte-ідентифікатора в функції і ID модуля забезпечує різницю між звичайним повідомленням і повідомленням 2ro рівня. Функція визначає тип повідомлення (об'єкту) ID модуля приймача.

Пристрої CANopen обмінюються даними в формі об'єктів. Профіль зв'язку CANopen визначає два різних типи об'єктів, а також кількість спеціальних об'єктів.

VIPA CAN-майстер підтримує наступні об'єкти:

- 40 PDO передачі (PDO зв'язку, PDO перерозподілу)
- 40 PDO приймання (PDO зв'язку, PDO перерозподілу)
- 2 стандартних SDO (1 сервер, 127 клієнт)
- 1 Аварійний об'єкт
- 1 Об'єкт мережевого керування NMT
- Контроль вузла
- Робочий цикл

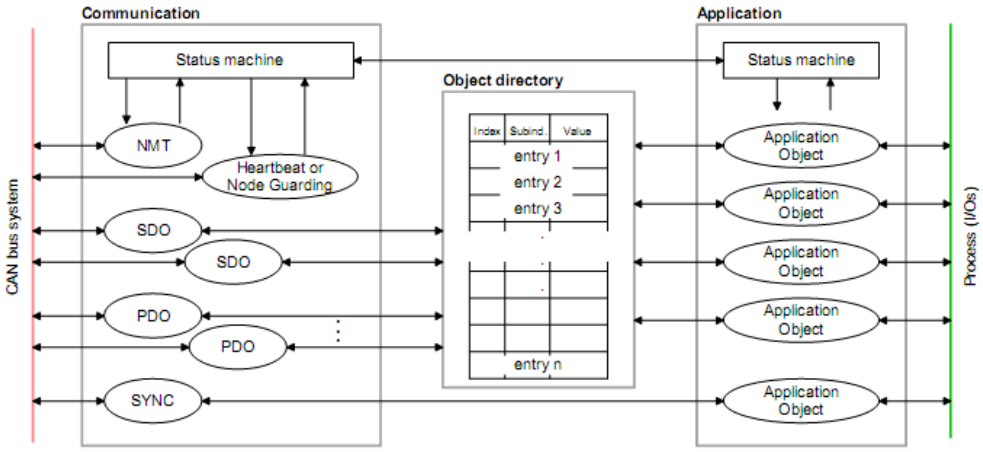


Примітка!

Точна структура і зміст даних усіх об'єктів описано в профілях CiA DS-301, DS-302, DS-401 і DS-405.

Модельна структура пристроїв

Пристрій CANopen може бути структуровано таким чином:



Зв'язок

Відповідає за передачу об’єктних даних по мережі CANopen.

Додаток

Об'єкти даних додатку містять вхідні і вихідні дані. У випадку помилки, спеціальний статусний механізм додатку переведе усі виходи в безпечний стан.

Об'єктну директорію організовано у вигляді 2^{-x} вимірної таблиці. Дані адресуються через індекси і підіндекси.

Об'єктна директорія

Ця об'єктна директорія містить всі доступні об'єктні дані (дані додатків + параметри) які впливають на поведінку зв'язку, додатку і стану пристрою.

PDO

У багатьох мережевих системах образ процесу передається в циклічному режимі. CANopen не має таких обмежень, оскільки підтримує більше можливостей передачі образу. Це обумовлено можливістю реалізації мульти-майстерної комунікації.

В CANopen дані процесу поділено на сегменти по макс. 8Byte. Ці сегменти називаються Процесними Об’єктами Даних (PDO). Кожен PDO представляє одну CAN-телеграму і ідентифікується по спеціальному CAN-ідентифікатору.

Для обміну даними процесу, VIPA CAN-майстер підтримує 80 PDO. Кожен PDO складається максимум з 8 байт даних. Передача PDO не перевіряється шляхом підтвердження, оскільки CAN протокол гарантує передачу даних.

Існує 40Tx PDO передачі для вхідних даних і 40Rx PDO отримання для вихідних даних.

PDO отримання (RxPDO) приймаються CAN-майстром і містять вхідні дані.

PDO передачі (TxPDO) відсилаються CAN-майстром і містять вихідні дані.

Призначення PDO вхідним чи вихідним даним відбувається через WinCoCT автоматично.

SDO

Для доступу до об'єктної директорії використовуються Сервісні Об'єкти Даних (SDO). SDO дає дозвіл на читання або запис в об'єктну директорію. SDO використовує специфікацію Протокол-Множинної-Передачі-Домену, яка детально описана в CAL-Layer-7-Protocol. Цей протокол дозволяє передавати дані будь-якої довжини. При необхідності, повідомлення діляться на кілька CAN-повідомлень з однаковим ідентифікатором (сегментація). SDO вимагає процедури підтвердження. Тому кожний прийом повідомлення проходить процедуру підтвердження.



Примітка!

Докладний опис SDO-телеграм можна знайти в стандарті CiA DS-301.

Далі описуються лише повідомлення про тривоги, які виникають при неправильних комунікаційних параметрах.

SFC 219 CAN_TLGR SDO запит до CAN майстра

Кожен CPU має інтегрований SFC 219. Це дозволяє CAN-майстру через програму PLC активувати SDO зчитування та запис.

Адресація до майстра здійснюється через положення підключення, а адресація до слейвів здійснюється через CAN-адреси. Дані процесу визначаються індексом і підіндексом. За допомогою SDO, при кожному сеансі доступу, передається максимум одне слово даних процесу.

SFC 219 містить наступні параметри:

Ім'я	Об'явлення	Тип	Коментар
Request	IN	BOOL	
Slot_Master	IN	BYTE	
NodeID	IN	BYTE	
Transfertyp	IN	BYTE	
Index	IN	DWORD	
Subindex	IN	DWORD	
CanOpenError	OUT	DWORD	
RetVal	OUT	WORD	
Busy	OUT	BOOL	
DataBuffer	IN_OUT	ANY	

Request Контрольований параметр: 1: Стартовий порядок в залежності від номера положення підключення.

Slot_Master 0: для адресації до інтегрованого-CAN майстра
1 ... 32: для адресації до автономного CAN-майстра System 200V.

NodeID Адресація до вузла CANopen (1 ... 127)

Transfer type 40h, 60h: Читання SDO
61h: Запис SDO (невизначена довжина)
23h: Запис SDO (1 DWORD)
2Bh: Запис SDO (1 WORD)
2Fh: Запис SDO (1 BYTE)

Index CANopen Індекс

Subindex CANopen Підіндекс

CanOpenError

Якщо помилки відсутні, CANopenError повертає значення 0.

У випадку помилки, CANopenError містить одне з наведених нижче повідомлень, які генеруються в CAN майстрі:

Код	Опис
0x05030000	Значення біту перемикання не чергується
0x05040000	Тайм-аут SDO-протоколу
0x05040001	Специфікатор команди клієнта/сервера є не дійсним чи невідомий
0x05040002	Невірний розмір блоку (лише режим блоків)
0x05040003	Невірний номер послідовності (лише режим блоків)
0x05040004	CRC-помилка (лише режим блоків)
0x05040005	Недостатньо пам'яті
0x06010000	Непідтримуваний доступ до об'єкту
0x06010001	Спроба читання об'єкту який можна лише записувати
0x06010002	Спроба запису об'єкту який можна лише читати
0x06020000	Об'єкт не існує в словнику об'єктів
0x06040041	Об'єкт не може бути зіставлений з PDO
0x06040042	Кількість і довжина об`єктів які будуть відображатися перевищать довжину PDO
0x06040043	Загальна несумісність параметрів
0x06040047	Загальна внутрішня несумісність в пристрої
0x06060000	Збій доступу через апаратну помилку
0x06070010	Тип даних не збігається, довжина сервісних параметрів не збігається
0x06070012	Тип даних не збігається, довжина сервісних параметрів занадто велика
0x06070013	Тип даних не збігається, довжина сервісних параметрів занадто коротка
0x06090011	Підіндекс відсутній
0x06090030	Діапазон значень параметра перевищено (тільки для запису)
0x06090031	Записане значення параметра занадто велике
0x06090032	Записане значення параметра занадто мале
0x06090036	Максимальне значення менше мінімального
0x08000000	Загальна помилка
0x08000020	Дані не можуть бути передані або збережені в додатку
0x08000021	Дані не можуть бути передані або збережені в додатку через місцеве керування
0x08000022	Дані не можуть бути передані або збережені в додатку через поточний стан пристрою
0x08000023	Динамічна генерація об'єктного словника дала збій або об'єктний словник відсутній (наприклад, об'єктний словник генерується з файлу, при цьому ця операція дає збій у зв'язку з помилкою файлу)

RetVal

Коли функція була виконана успішно, повертається значення, що містить актуальну довжину даних відповіді: 1: BYTE, 2: WORD, 4: DWORD.

Якщо під час обробки функції відбувається помилка, назад повертається значення, що містить код цієї помилки.

Код	Опис
F021h	Невірна адреса слейва (параметр дорівнює 0 або вище 127)
F022h	Невірний тип передачі (значення не рівне 60h, 61h)
F023h	Невірна довжина даних (буфер даних занадто малий, при SDO-доступі на читання він повинен бути не менше 4Byte, при SDO-доступі на запис - 1Byte, 2 Byte або 4Byte)
F024h	SFC не підтримується
F025h	Буфер запису в CANopen-майстрі переповнено, в цей час задача не може бути виконана
F026h	Буфер читання в CANopen-майстрі переповнено, в цей час задача не може бути виконана
F027h	Процедура SDO-читання чи запису повернула невірну відповідь, дивіться коди помилок CANopen
F028h	SDO-таймаут (не знайдено жодного CANopen-учасника з таким ID вузла)

Busy

BUSY = 1: Операція читання/запису не закінчена.

DataBuffer

SFC-комунікаційна область даних. Задайте тут будь-який покажчик типу Byte.

Зчитування SDO: Області призначення для SDO-даних, які читалися.

Запис SDO: Вихідні області для SDO-даних, які були записані.



Примітка!

Якщо SDO-запит був оброблений без помилок, RetVal містить довжину достовірних даних відповіді в байтах 1, 2 або 4, а CanOpenError - значення 0.

Об’єктна директорія

Структура

Об’єктна директорія CANopen містить всі необхідні для мережевого розподілювача об'єкти. Кожний запис в об’єктній директорії виділяється 16Bit-ним індексом.

Якщо об'єкт міститься в кількох компонентах (наприклад, об'єкт типу Масив або Запис), компоненти виділяються через 8Bit-ний підіндекс. Ім'я об'єкта описує його функцію. Атрибут типу даних визначає тип даних запису.

Атрибут доступу визначає можливість читання, запису або читання і запису.

Об'єктна директорія розділена на наступні 3 частини:

Специфічна область комунікації профілю (0x1000 – 0x1FFF)

Ця область містить опис всіх відповідних параметрів зв'язку

0x1000 – 0x1011	Загальні комунікаційні параметри (наприклад, ім’я пристрою)
0x1400 – 0x1427	Комунікаційні параметри (наприклад, ідентифікатор) отриманих PDO
0x1600 – 0x1627	Параметри відображення отриманих PDO Розподіл параметрів виконано у вигляді перехресних зв’язків з об'єктами додатку, які відображаються в PDO, і містить ширину даних залежних об'єктів.
0x1800 – 0x1827 0x1A00 – 0x1A27	Комунікаційні і розподілені параметри переданих PDO

Специфічна область профілю виробника (0x2000 – 0x5FFF)

Тут ви знайдете інформацію виробника. CAN-майстер від VIPA не містить жодних записів виробника.

Стандартна область профілю пристрою (0x6000 – 0x9FFF)

Ця область містить об'єкти профілю пристрою, згідно з DS-401.



Примітка!

Таблиця об'єктів адаптована відповідно до стандарту CiA. Нижче в таблицях наводяться дані об’єктів.

Під таблицями подано пояснення до їх записів.

Огляд об’єктної директорії

Індекс	Вміст об’єкту
1000h	Тип пристрою
1001h	Регістр помилок
1005h	COB-ID SYNC
1006h	Період комунікаційного циклу
1007h	Роздір вікна синхронізації
1008h	Апаратна версія виробника
1009h	Апаратна версія
100Ah	Програмна версія
100Ch	Контрольний час
100Dh	Часовий коефіцієнт напрацювання
1016h	Споживацький час роботи
1017h	Виробничий час роботи
1018h	Об’єкт ідентичності
1400h - 1427h	Комунікаційний параметр отриманих PDO
1600h - 1627h	Розподіл параметрів отриманих PDO
1800h - 1827h	Комунікаційний параметр переданих PDO
1A00h - 1A27h	Розподіл параметрів переданих PDO
1F22h	Стиснутий DCF
1F25h	Конфігурація повідомлень
1F80h	NMT-ініціація
1F81h	Призначення слейва
1F82h	NMT-запит
1F83h	Запит контролю
6000h	Масив DI8-Bit (див. DS 401)
6100h	Масив DI16-Bit (див. DS 401)
6120h	Масив DI32Bit (див. DS 401)
6200h	Масив DO8-Bit (див. DS 401)
6300h	Масив DO16-Bit (див. DS 401)
6320h	Масив DO32-Bit (див. DS 401)
A040h	Динамічний беззнаковий 8 Input
A100h	Динамічний беззнаковий 16 Input
A200h	Динамічний беззнаковий 32 Input
A4400h	Динамічний беззнаковий 64 Input
A4C0h	Динамічний беззнаковий 8 Output
A580h	Динамічний беззнаковий 16 Output
A680h	Динамічний беззнаковий 32 Output
A8C0h	Динамічний беззнаковий 64 Output

Тип пристрою

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0x1000	0	Тип пристрою	Unsigned32	ro	N	0x00050191	Декларування типу пристрою

Значення 32Bit розділено на два поля по 16Bit:

MSB	LSB
Додаткова інформація по пристрою	**номер профілю**
0000 0000 0000 wxyz (bit)	405dec=0x0195

"Додаткова інформація" містить дані, що відносяться до типів сигналів I/O пристрою:

z=1 дискретний вхід
y=1 дискретний вихід
x=1 аналоговий вхід
w=1 аналоговий вихід

Регістр помилок

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0x1001	0	Регістр помилок	Unsigned8	ro	Y	0x00	Регістр помилок

Bit 7							Bit 0
ManSpec	резерв	резерв	Comm.	резерв	резерв	резерв	Generic

ManSpec.: Специфічна помилка виробника, визначена для об'єкту 0x1003.
Comm.: Помилка комунікації (забиття CAN)
Generic: Поява неспецифічної помилки (флаг зводиться при кожному повідомленні про помилку)

SYNC-ідентифікатор

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0x1005	0	COB-Id повідомлення синхронізації	Unsigned32	ro	N	0x80000080	Ідентифікатор SYNC повідомлення

Молодші 11Bit з 32 Bit значення містять ідентифікатор (0x80 = 128dez), при цьому MSBit вказує, чи пристрій отримав телеграму SYNC (1) чи ні (0).

Увага: На відміну від PDO-ідентифікаторів, поява MSB вказує, що цей ідентифікатор має відношення до вузла.

SYNC-інтервал

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0x1006	0	Період комунікаційного циклу	Unsigned32	rw	N	0x00000000	Максимальна довжина SYNC-інтервалу в µs

Якщо тут встановлено значення відмінне від нуля, і протягом заданого часу синхронізації PDO, не отримано SYNC-телеграму, то майстер переходить в стан помилки.

Розмір вікна синхронізації

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0x1007	0	Розмір вікна синхронізації	Unsigned32	rw	N	0x00000000	Містить розмір часового вікна для синхронізації PDO в µs

Ім'я пристрою

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0x1008	0	Виробниче ім'я пристрою	Visible string	ro	N		Ім'я пристрою шинного розподілювача

VIPA 21x-2CM02

Так як повернене значення більше, ніж 4Byte, для передачі використовується сегментований SDO-протокол.

Апаратна версія

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0x1009	0	Апаратна версія виробника	Visible string	ro	N		Номер апаратної версії шинного розподілювача

1.00

Так як повернене значення більше, ніж 4Byte, для передачі використовується сегментований SDO-протокол.

Програмна версія

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0x100A	0	Програмна версія виробника	Visible string	ro	N		Номер версії прошивки програмного забезпечення CANopen

1.xx

Так як повернене значення більше, ніж 4Byte, для передачі використовується сегментований SDO-протокол.

Контрольний час

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0x100C	0	Контрольний час [ms]	Unsigned16	rw	N	0x0000	Інтервал між двома контрольними телеграмами. Встановлюється NMT-майстром чи конфігураційним інструментом

Часовий коефіцієнт напрацювання

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0x100D	0	Часовий коефіцієнт напрацювання	Unsigned8	rw	N	0x00	Часовий коефіцієнт напрацювання x контрольний час = час напрацювання (контрольний час)

Якщо протягом часу напрацювання не отримано контрольної телеграми, вузол переходить в стан помилки. Якщо часовий коефіцієнт напрацювання та/або контрольний час = 0, то вузол не контролює час напрацювання, але самостійно може перебувати під наглядом майстра (контроль вузла).

Споживацький час роботи

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0x1016	0	Споживацький час роботи	Unsigned8	ro	N	0x05	Кількість записів
	1…127		Unsigned32	rw	N	0x00000000	Споживацький час роботи

Bit	31-24	23-16	15-0
Значення	резерв	ID вузла	Час роботи
Кодується як	Unsigned8	Unsigned8	Unsigned16

Як тільки ви спробуєте для однакових ID вузла налаштувати споживацький час роботи не рівний нулю, вузол виконає переривання завантаження SDO і згенерує код помилки 0604 0043h.

Виробничий час роботи

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0x1017	0	Виробничий час роботи	Unsigned16	rw	N	0x0000	Визначення циклу виробничого часу роботи в ms

Об'єкт ідентичності

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0x1018	0	Об'єкт ідентичності	Unsigned16	ro	N	0x04	Містить загальну інформацію про пристрій (кількість записів)
		ID виробника	Unsigned32	ro	N	0xAFFEAFFE	ID виробника
		Код продукту	Unsigned32	ro	N	0x2142CA02	Код продукту
		Номер ревізії	Unsigned32	ro	N		Номер ревізії
		Серійний номер	Unsigned32	ro	N		Серійний номер

Комунікаційний параметр RxPDO

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0x1400 ... 0x1427	0	Кількість записів	Unsigned8	ro	N	0x02	Комунікаційний параметр першого отриманого PDO Підіндекс 0: кількість послідовних параметрів
	1	COB-ID	Unsigned32	rw	N	0xC0000200 + NODE_ID	COB-ID RxPDO1
	2	Тип передачі	Unsigned8	rw	N	0xFF	Тип передачі PDO

Підіндекс 1 (COB-ID): молодші 11Bit з 32Bit значення (Bit 0-10) містять ідентифікатор CAN. MSBit (Bit 31) показує чи PDO активний (1) чи ні (0), Bit 30 показує чи RTR доступ до цього PDO дозволено (0) чи ні (1).

Підіндекс 2 містить тип передачі.

Розподіл RxPDO

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0x1600 ... 0x1627	0	Кількість записів	Unsigned8	rw	N	0x01	Розподіл параметрів першого отриманого PDO; підіндекс 0: кількість розподілених об'єктів
	1	1й розподілений об'єкт	Unsigned32	rw	N	0x62000108	(2 Byte індекс, 1 Byte підіндекс, 1 Byte bit-ширина)
	2	2й розподілений об'єкт	Unsigned32	rw	N	0x62000208	(2 Byte індекс, 1 Byte підіндекс, 1 Byte bit-ширина)
	…	…	…	…	…	…	…
	8	8й розподілений об'єкт	Unsigned32	rw	N	0x62000808	(2 Byte індекс, 1 Byte підіндекс, 1 Byte bit-ширина)

Залежно від підключених модулів, прийняті PDO автоматично отримують розподілене значення.

Комунікаційний параметр TxPDO1

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0x1800 ... 0x1827	0	Кількість записів	Unsigned8	rw	N	0x05	Комунікаційний параметр першого переданого PDO, підіндекс 0: кількість послідовних параметрів
	1	COB-ID	Unsigned32	rw	N	0x80000180+ NODE_ID	COB-ID TxPDO1
	2	Тип передачі	Unsigned8	rw	N	0xFF	Тип передачі PDO
	3	Час заборони	Unsigned16	rw	N	0x0000	Повтор затримок [значення x 100 µs]
	5	Час події	Unsigned16	rw	N	0x0000	Таймер подій [значення x 1 µs]

Підіндекс 1 (COB-ID): молодші 11Bit з 32Bit значення (Bit 0-10) містять ідентифікатор CAN. MSBit (Bit 31) показує чи PDO активний (1) чи ні (0), Bit 30 показує чи RTR доступ до цього PDO дозволено (0) чи ні (1).

Підіндекс 2 містить тип передачі, підіндекс 3 відповідає за повтор затримок між двома рівними PDO. Якщо таймер подій має значення не рівне 0, то PDO не передасться.

Якщо активний "таймер заборони", подія відкладається.

Розподіл TxPDO1

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0x1A00 ... 0x1A27	0	Кількість записів	Unsigned8	rw	N	Залежно від встановлених компонентів	Розподіл параметрів першого переданого PDO; підіндекс 0: кількість розподілених об'єктів
	1	1й розподілений об'єкт	Unsigned32	rw	N	0x60000108	(2 Byte індекс, 1 Byte підіндекс, 1 Byte bit-ширина)
	2	2й розподілений об'єкт	Unsigned32	rw	N	0x60000208	(2 Byte індекс, 1 Byte підіндекс, 1 Byte bit-ширина)
	…	…	…	…	…	…	…
	8	8й розподілений об'єкт	Unsigned32	rw	N	0x60000808	(2 Byte індекс, 1 Byte підіндекс, 1 Byte bit-ширина)

Залежно від підключених модулів, відправлені PDO автоматично отримують розподілене значення.

Стиснутий DCF

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0x1F22	Масив	Стиснутий DCF	Domain	rw	N		

Цей об'єкт необхідний для конфігураційного менеджера. Стиснутий DCF – це укорочена форма DCF (**Ф**айл **К**онфігурації **П**ристрою).

Конфігурація повідомлень

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0x1F25	Масив	Конфігурування слейва	Unsigned32	rw	N	0x00000000	

Через цей параметр, можна ініціювати передачу конфігураційним менеджером збереженої конфігурації в мережу.

Конфігурування визначеного вузла може бути ініційоване у будь-який час за допомогою індексу 0x1F25.

Підіндекс 0 має значення 128.

Підіндекс x (при x=1…127): Починає переконфігурацію вузлів через його ID.

Підіндекс 128: переконфігурація усіх вузлів.

Наприклад: Якщо ви хочете розпочати конфігурування вузла 2, і при цьому доступні конфігураційні дані, в об'єкт 1F25h підіндекс 2 Ви повинні записати значення 0x666E6F63 (ASCII = "conf").

NMT-ініціація

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0x1F80	0x00	NMT-ініціація	Unsigned32	rw	N	0x00000000	

Визначення пристрою як NMT-майстер.

Bit	Значення
Bit 0	0: Пристрій НЕ являється NMT-майстром. Усі інші біти повинні ігноруватись. Об'єкти мережевого списку повинні ігноруватись. 1: Пристрій являється NMT-майстром.
Bit 1	0: Запуск тільки явно визначених слейвів. 1: Після завантаження виконається сервіс NMT Start Remote Node
Bit 2..31	Зарезервовано CiA, завжди 0

Призначення слейва

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0x1F81	0x00	Призначення слейва	Unsigned32	rw	N	0x00000000	

Визначення вузлів які контролюються майстром. Для кожного вузла необхідно задати один запис.

Підіндекс 0 має значення 127. Кожен другий підіндекс відповідає ID-вузла.

Byte	Bit	Значення
Byte 0	Bit 0	0: Вузол з таким ID не являється слейвом 1: Вузол з таким ID являється слейвом. Після конфігурування (за допомогою конфігураційного менеджера) Вузол буде переведено в стан «Робочий».
	Bit 1	0: При Події Контролю Помилки чи іншому вияві завантаження слейва, буде проінформовано додаток. 1: При Події Контролю Помилки чи іншому вияві завантаження слейва, буде проінформовано додаток і автоматично запуститься Служба Контролю Помилок.
	Bit 2	0: При Події Контролю Помилки чи іншому вияві завантаження слейва він не буде автоматично конфігурувати та запускати його. 1: При Події Контролю Помилки чи іншому вияві завантаження слейва, буде запущено процес Start Boot Slave.
	Bit 3..7	Зарезервовано CiA, завжди 0
Byte 1		8 Bit значення для RetryFactor
Byte 2, 3		16 Bit значення для GuardTime

NMT-запит

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0x1F82	0x00	NMT-запит	Unsigned32	rw	N	0x00000000	

Якщо автоматичний запуск повного стеку не бажаний, функціональні можливості

• зміни статусу

• початку контролю

• конфігурування через CMT

можуть запускатись вибірково для кожного вузла по запиту. Запит завжди відбувається через об'єкти в об'єктній директорії.

Перемикання станів зв'язку всіх вузлів в мережі (у тому числі

Місцевих слейвів) виконується через запис 1F82h в локальній об'єктній директорії:

Підіндекс 0 має значення 128.
Підіндекс x (при x=1..127): Ініціація NMT-сервісу для вузлів через їх IDx.
Підіндекс 128: Ініціація NMT-сервісу для усіх вузлів.

При записі потрібний стан задається в якості значення.

Стан	Значення
Підготовка	4
Робота	5
Скидання вузла	6
Скидання комунікації	7
Перед операційний	127

Запит контролю

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0x1F83	0x00	Запит контролю	Unsigned32	rw	N	0x00000000	

Підіндекс 0 має значення 128.
Підіндекс x (при x=1..127): Ініціація NMT-сервісу для слейвів через їх IDx вузлів.

Значення	Доступ запису	Доступ читання
1	Початок контролю	Слейв контролюється
0	Стоп контролю	Слейв не контролюється

Підіндекс 128: Запит Старт/Стоп Контролю для усіх вузлів.

8Bit DI

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0x6000	0x00	8Bit блок дискретних входів	Unsigned8	ro	N	0x01	Кількість доступних 8Bit блоків дискретних входів
	0x01	1й вхідний блок	Unsigned8	ro	Y		1й дискретний вхідний блок
	…	…	…	…	…	…	…
	0x40	64й вхідний блок	Unsigned8	ro	Y		64й дискретний вхідний блок

16Bit DI

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0x6100	0x00	16Bit блок дискретних входів	Unsigned8	ro	N	Залежно від встановлених компонентів	Кількість доступних 16Bit блоків дискретних входів
	0x01	1й вхідний блок	Unsigned16	ro	N		1й дискретний вхідний блок
	…	…	…	…	…	…	…
	0x20	32й вхідний блок	Unsigned16	ro	N		32й дискретний вхідний блок

32Bit DI

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0x6120	0x00	32Bit блок дискретних входів	Unsigned8	ro	N	Залежно від встановлених компонентів	Кількість доступних 32Bit блоків дискретних входів
	0x01	1й вхідний блок	Unsigned32	ro	N		1й дискретний вхідний блок
	…	…	…	…	…	…	…
	0x10	16й вхідний блок	Unsigned32	ro	N		16й дискретний вхідний блок

8Bit DO

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0x6200	0x00	8Bit блок дискретних виходів	Unsigned8	ro	N	0x01	Кількість доступних 8Bit блоків дискретних виходів
	0x01	1й вихідний блок	Unsigned8	rw	Y		1й дискретний вихідний блок
	…	…	…	…	…	…	…
	0x40	64й вихідний блок	Unsigned8	rw	Y		64й дискретний вихідний блок

16Bit DO

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0x6300	0x00	16Bit блок дискретних виходів	Unsigned8	ro	N	Залежно від встановлених компонентів	Кількість доступних 16Bit блоків дискретних виходів
	0x01	1й вихідний блок	Unsigned16	rw	N		1й дискретний вихідний блок
	…	…	…	…	…	…	…
	0x20	32й вихідний блок	Unsigned16	rw	N		32й дискретний вихідний блок

32Bit DO

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0x6320	0x00	32Bit блок дискретних виходів	Unsigned8	ro	N	Залежно від встановлених компонентів	Кількість доступних 32Bit блоків дискретних виходів
	0x01	1й вихідний блок	Unsigned32	rw	N		1й дискретний вихідний блок
	…	…	…	…	…	…	…
	0x10	16й вихідний блок	Unsigned32	rw	N		16й дискретний вихідний блок

8Bit Мережеві вхідні змінні

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0xA040	0x00	8Bit блок дискретних входів	Unsigned8	ro	N	0x01	Кількість доступних 8Bit блоків дискретних входів
	0x01	1й вхідний блок	Unsigned8	ro	Y		1й дискретний вхідний блок
	…	…	…	…	…	…	…
	0x140	320й вхідний блок	Unsigned8	ro	Y		320й дискретний вхідний блок

16Bit Мережеві вхідні змінні

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0xA100	0x00	16Bit блок дискретних входів	Unsigned8	ro	N	Залежно від встановлених компонентів	Кількість доступних 16Bit блоків дискретних входів
	0x01	1й вхідний блок	Unsigned16	ro	N		1й дискретний вхідний блок
	…	…	…	…	…	…	…
	0xA0	160й вхідний блок	Unsigned16	ro	N		160й дискретний вхідний блок

32Bit Мережеві вхідні змінні

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0xA200	0x00	32Bit блок дискретних входів	Unsigned8	ro	N	Залежно від встановлених компонентів	Кількість доступних 32Bit блоків дискретних входів
	0x01	1й вхідний блок	Unsigned32	ro	N		1й дискретний вхідний блок
	…	…	…	…	…	…	…
	0x50	80й вхідний блок	Unsigned32	ro	N		80й дискретний вхідний блок

64Bit Мережеві вхідні змінні

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0xA440	0x00	64Bit блок дискретних входів	Unsigned8	ro	N	Залежно від встановлених компонентів	Кількість доступних 64Bit блоків дискретних входів
	0x01	1й вхідний блок	Unsigned32	ro	N		1й дискретний вхідний блок
	…	…	…	…	…	…	…
	0x28	40й вхідний блок	Unsigned32	ro	N		40й дискретний вхідний блок

8Bit Мережеві вихідні змінні

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0xA400	0x00	8Bit блок дискретних виходів	Unsigned8	ro	N	0x01	Кількість доступних 8Bit блоків дискретних виходів
	0x01	1й вихідний блок	Unsigned8	rw	Y		1й дискретний вихідний блок
	…	…	…	…	…	…	…
	0x140	320й вихідний блок	Unsigned8	rw	Y		320й дискретний вихідний блок

16Bit Мережеві вихідні змінні

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0xA580	0x00	16Bit блок дискретних виходів	Unsigned8	ro	N	Залежно від встановлених компонентів	Кількість доступних 16Bit блоків дискретних виходів
	0x01	1й вихідний блок	Unsigned16	rw	N		1й дискретний вихідний блок
	…	…	…	…	…	…	…
	0xA0	160й вихідний блок	Unsigned16	rw	N		160й дискретний вихідний блок

32Bit Мережеві вихідні змінні

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0xA680	0x00	32Bit блок дискретних виходів	Unsigned8	ro	N	Залежно від встановлених компонентів	Кількість доступних 32Bit блоків дискретних виходів
	0x01	1й вихідний блок	Unsigned32	rw	N		1й дискретний вихідний блок
	…	…	…	…	…	…	…
	0x50	80й вихідний блок	Unsigned32	rw	N		80й дискретний вихідний блок

64Bit Мережеві вихідні змінні

Індекс	Під індекс	Ім'я	Тип	Атр.	Розп.	Типове значення	Значення
0xA8C0	0x00	64Bit блок дискретних виходів	Unsigned8	ro	N	Залежно від встановлених компонентів	Кількість доступних 64Bit блоків дискретних виходів
	0x01	1й вихідний блок	Unsigned32	rw	N		1й дискретний вихідний блок
	…	…	…	…	…	…	…
	0x50	40й вихідний блок	Unsigned32	rw	N		40й дискретний вихідний блок