

## ***How-To-Do***

### **Управление преобразователем частоты YASKAWA по сети Modbus RTU**

## 1 Обзор

### Используемое оборудование и ПО

- SPEED7 Studio с версии V1.7.1 или
- Siemens SIMATIC Manager с версии V5.5 SP2 и библиотека *Simple Motion Control* или
- Siemens TIA Portal V14 и библиотека *Simple Motion Control*.
- Модуль ЦПУ серий MICRO или SLIO с последовательным интерфейсом, такой как CPU M13-CCF0000 или CPU 013-CCF0R00.
- Преобразователь частоты V1000 с последовательным интерфейсом и соответствующий электродвигатель.

### Последовательность конфигурирования

1. ➔ Установка параметров преобразователя частоты.
  - Установка параметров осуществляется с помощью программного обеспечения *Drive Wizard+*.
2. ➔ Конфигурирование аппаратных средств в среде VIPA *SPEED7 Studio*, Siemens SIMATIC Manager или Siemens TIA Portal.
  - Конфигурирование модуля ЦПУ.
3. ➔ Программирование в среде VIPA *SPEED7 Studio*, Siemens SIMATIC Manager или Siemens TIA Portal.
  - Запараметрируйте функциональный блок для последовательной связи.
  - Запараметрируйте функциональный блок для каждого ведомого устройства Modbus.
  - Запараметрируйте функциональный блок для всех ведомых устройств Modbus.
  - Запараметрируйте функциональный блок с коммуникационными данными для всех ведомых устройств Modbus.
  - Запараметрируйте функциональный блок для диспетчера обмена.
  - Запараметрируйте функциональный блок для инициализации преобразователя частоты.
  - Запараметрируйте функциональный блок для режима движения.

## 2 Установка параметров преобразователя частоты



### ВНИМАНИЕ!

Перед вводом в эксплуатацию необходимо адаптировать преобразователь частоты применительно к решаемой задаче с помощью программного обеспечения *Drive Wizard+*! Дополнительную информацию можно найти в руководстве пользователя для используемого преобразователя частоты.

В следующей таблице приведены все параметры, которые не соответствуют значениям по умолчанию. Эти параметры должны быть заданы с помощью *Drive Wizard+* для обеспечения их соответствия библиотеке *Simple Motion Control*.

№	Обозначение	Диапазон значений	Значение для библиотеки <i>Simple Motion Control</i>
H5-01	Адрес ведомого устройства для преобразователя частоты	00h, 20h	По умолчанию значение адреса ведомого устройства равно 1Fh. Обратите внимание, что сетевые адреса устройств не должны повторяться!
H5-02	Скорость передачи данных для сети MEMOBUS / Modbus	0, 1, 2, ..., 8	■ 3: 9600 бит/с
H5-03	Режим проверки по четности для обмена по протоколу MEMOBUS/Modbus	0, 1, 2	■ 0: без проверки по четности

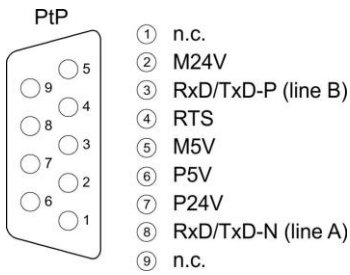
№	Обозначение	Диапазон значений	Значение для библиотеки <i>Simple Motion Control</i>
H5-04	Метод остановки двигателя в случае возникновения коммуникационной ошибки (ошибка CE)	0, 1, 2, 3	■ 3: Продолжение работы с формированием сигнала тревоги
H5-05	Обнаружение коммуникационной ошибки (ошибка CE)	0, 1	■ 1: Включено. Если соединение прерывается на более 2 секунд (настраивается через H2-09), формируется ошибка CE.
H5-06	Время ожидания между приемом и отправкой данных из преобразователя частоты	5 ... 65 мс	■ 5 мс
H5-07	Управление RTS (Request to send)	0, 1	■ 1: 1: Включено. RTS активируется только при передаче (RS-485 или RS-422 и <i>многоточечный</i> )
H5-09	Время, в течение которого обнаруживается ошибка связи (ошибка CE).	0,0 ... 10,0 с	■ 2 с
H5-10	Размер шага (разрешение) для регистра MEMOBUS / Modbus 0025h	0, 1	По умолчанию значение шага приращений равно 0,1 В (0). ■ 0: Приращение 0,1 В ■ 1: Приращение 1 В
H5-11	Функция ENTER для соединений	0, 1	■ 1: Команда ENTER не требуется
H5-12	Выбор метода команды запуска	0, 1	■ 1: Run/Stop
V1-01	Источник опорной частоты 1	0, 1, 2, 3, 4	■ 2: Сеть MEMOBUS/Modbus
V1-02	Источник команды пуска 1	0, 1, 2, 3	■ 2: Сеть MEMOBUS/Modbus
V1-15	Источник опорной частоты 2	0, 1, 2, 3, 4	■ 2: Сеть MEMOBUS/Modbus
V1-16	Источник команды пуска 2	0, 1, 2, 3	■ 2: Сеть MEMOBUS/Modbus



*Чтобы все настройки вступили в действие, необходимо после параметризации перезапустить преобразователь частоты!*

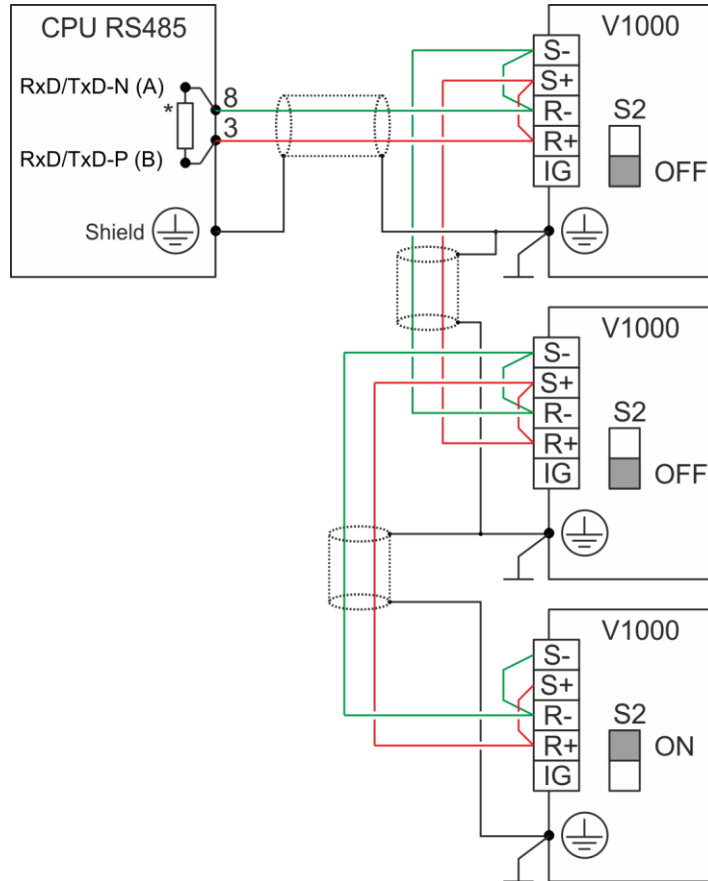
### 3 Подключение

#### Кабельная разводка сети RS-485



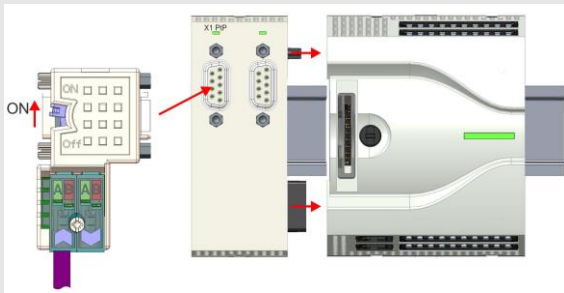
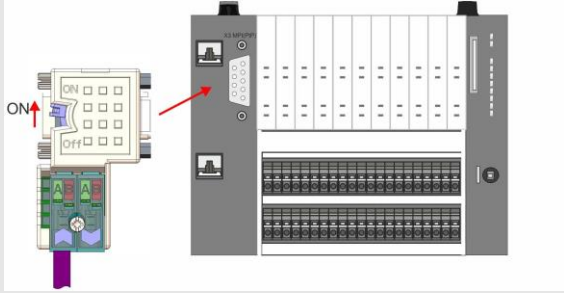
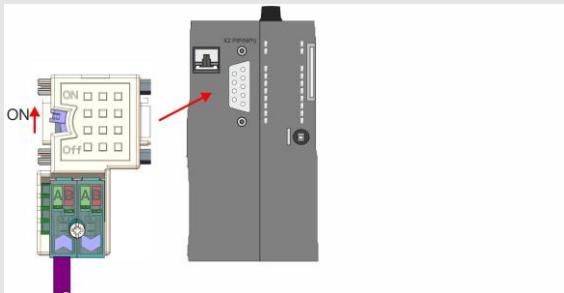
На следующем рисунке показано подключение преобразователей частоты V1000 к сети RS-485. Отдельные преобразователи частоты соединяются между собой кабелем PROFIBUS и с использованием соединителя PROFIBUS подключаются к порту PtP (Point-to-Point) модуля ЦПУ.

- Для всех подключенных преобразователей частоты значение параметра H5-07 должно быть установлено в 1.
- Линия связи должна иметь на обоих концах согласующие резисторы (терминальные). Для активации терминального резистора в преобразователе частоты переведите его переключатель S2 в положение 'ON'.



- \*) Для обеспечения надежного обмена данными установите на стороне ЦПУ согласующий резистор с номинальным сопротивлением примерно 120 Ом, в качестве которого можно использовать терминальный резистор, встроенный в соединитель PROFIBUS компании VIPA.
- Никогда не подключайте экран кабеля к контакту 5 (цель M5V) соединителя PROFIBUS во избежание вывода из строя последовательных интерфейсов устройств выравнивающими токами!

**Подключение процессорного модуля**

Процессорный модуль	Подключение	Примечание
<p>MICRO CPU M13C</p>		<ul style="list-style-type: none"> <li>■ Для организации связи PtP требуется дополнительный модуль расширения EM M09.</li> <li>■ В модуле расширения используется порт X1: PtP (RS422/485) с фиксированным назначением контактов.</li> <li>■ Для подключения кабельной линии к процессорному модулю используйте соединитель PROFIBUS производства компании VIPA.</li> <li>■ Активируйте в соединителе PROFIBUS терминальный резистор.</li> <li>■ После подачи питания и короткой процедуры запуска процессорный модуль готов для обмена данными через PtP-интерфейс.</li> </ul>
<p>SLIO CPU 013C</p>		<ul style="list-style-type: none"> <li>■ В процессорном модуле используется встроенный порт X3 MPI(PtP) с фиксированным назначением контактов.</li> <li>■ Для подключения кабельной линии к процессорному модулю используйте соединитель PROFIBUS производства компании VIPA.</li> <li>■ Активируйте в соединителе PROFIBUS терминальный резистор.</li> <li>■ После подачи питания и короткой процедуры запуска или после выполнения полного сброса последовательный порт имеет функционал MPI. Активирование функционала PtP можно выполнить в аппаратном конфигураторе.</li> </ul> <p>↪ <i>Раздел 4 'Использование VIPA SPEED7 Studio' на стр. 6.</i></p> <p>↪ <i>Раздел 5 'Использование Siemens SIMATIC Manager' на стр. 21.</i></p> <p>↪ <i>Раздел 6 'Использование Siemens TIA Portal' на стр. 35.</i></p>
<p>SLIO CPU 014 ... 017</p>		<ul style="list-style-type: none"> <li>■ В процессорном модуле используется встроенный порт X2 PtP (MPI), который по умолчанию настроен на выполнение функционала PtP (Point-to-Point).</li> <li>■ Для подключения кабельной линии к процессорному модулю используйте соединитель PROFIBUS производства компании VIPA.</li> <li>■ Активируйте в соединителе PROFIBUS терминальный резистор.</li> <li>■ После подачи питания и короткой процедуры запуска процессорный модуль готов для обмена данными через PtP-интерфейс.</li> </ul>

## 4 Использование VIPA SPEED7 Studio

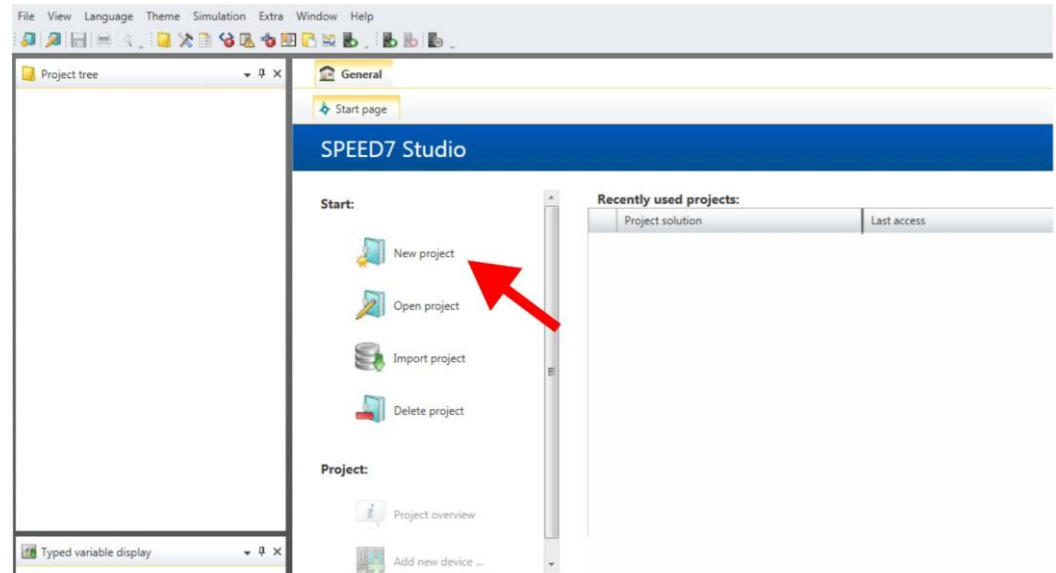
### 4.1 Конфигурирование аппаратных средств

#### 4.1.1 Конфигурирование контроллера серии MICRO

##### Добавление модуля ЦПУ в проект

Используйте для конфигурирования *SPEED7 Studio V1.7.1* и выше.

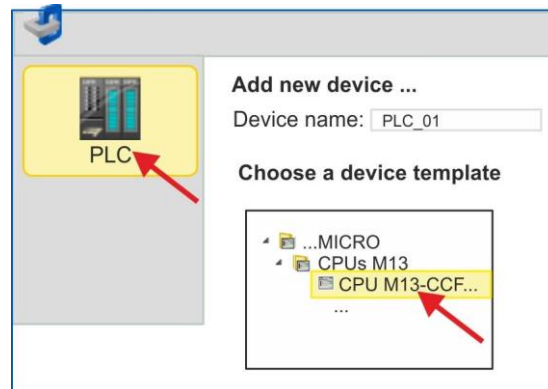
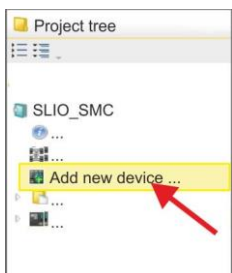
1. ▶ Запустите *SPEED7 Studio*.



2. ▶ Создайте новый проект на стартовой странице с помощью команды 'New project' и задайте ему имя с помощью 'Project name'.

⇒ Новый проект создается и отображается в окне 'Devices and networking'.

3. ▶ Кликните в дереве проекта *Project tree* по 'Add new device ...'.



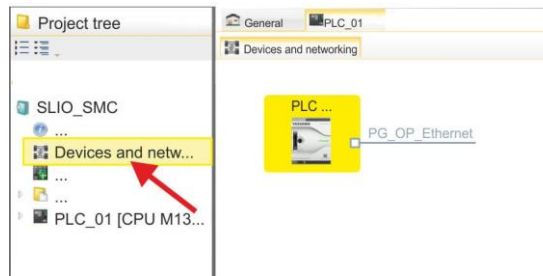
⇒ Откроется диалоговое окно выбора устройства.

4. ▶ Выберите из 'Device templates' процессорный модуль MICRO CPU M13-CCF0000 и кликните по [OK].

⇒ ЦПУ будет добавлен в разделы 'Devices and networking' и откроется окно 'Device configuration'.

**Конфигурирование порта Ethernet PG/OP**

1. ➤ Кликните в дереве проекта *Project tree* по *'Devices and networking'*.  
⇒ Вы получите графическое представление объекта используемого ЦПУ.

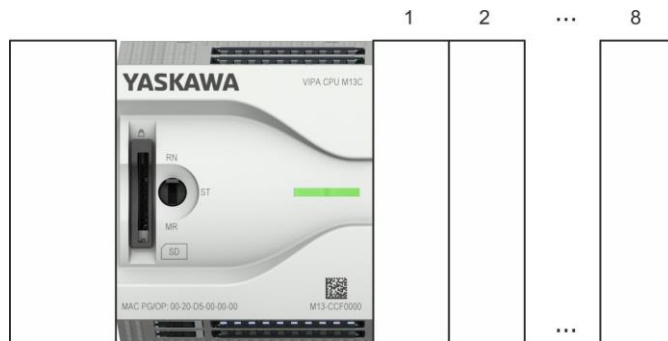


2. ➤ Кликните по изображению сети *'PG\_OP\_Ethernet'*.
3. ➤ Выберите *'Context menu → Interface properties'*.  
⇒ Откроется диалоговое окно. В нем необходимо ввести IP-адрес для порта Ethernet PG/OP. Предварительно необходимо получить у администратора сети допустимый IP-адрес.
4. ➤ Подтвердите нажатием по [OK].  
⇒ Данные IP-адреса сохраняются в проекте и будут отображены в окне *'Local components'* раздела *'Devices and networking'*.

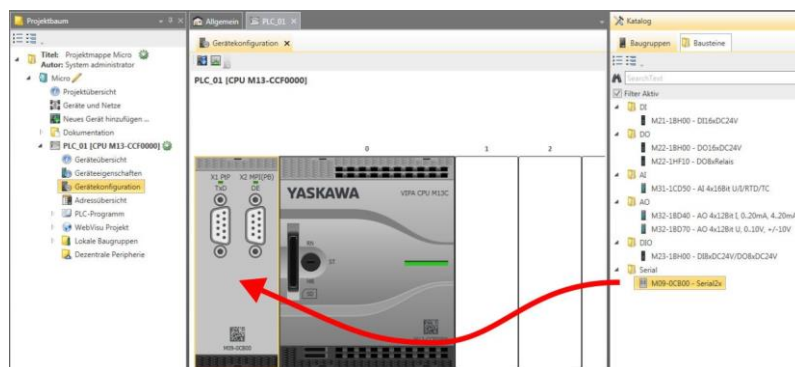
После загрузки проекта в используемый ЦПУ можно получить доступ к устройству через порт Ethernet PG/OP с использованием заданного для него IP-адреса.

**Активирование функциональности PtP**

1. ➤ В менеджере проекта *Project tree* кликните по *'PLC..CPU M13... → Device configuration'*.  
⇒ Откроется окно *Device configuration'*.



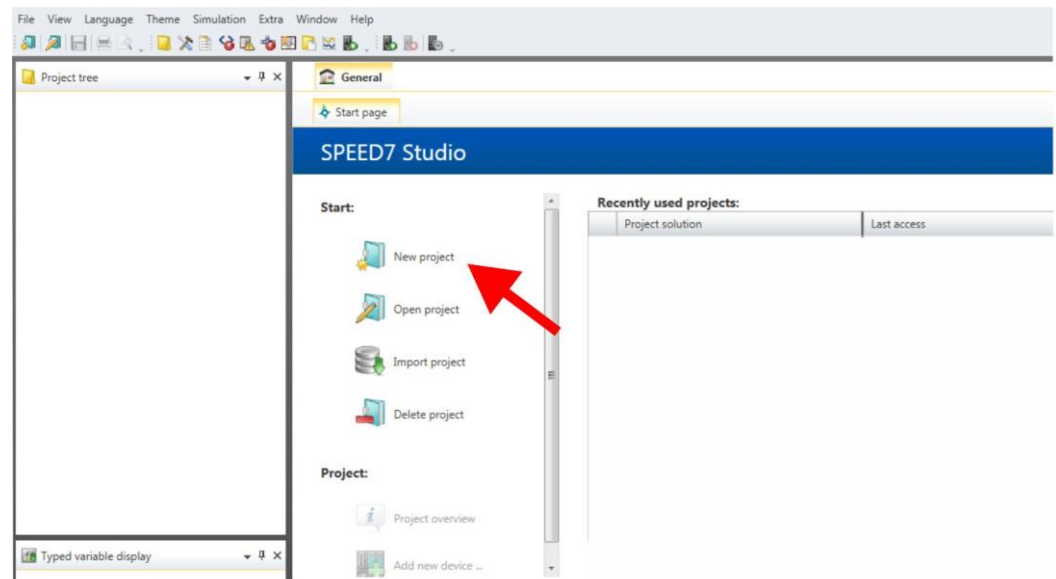
2. ➤ В разделе *'Components'* вкладки *'Catalog'* откройте библиотеку *'Serial'* и перетащите модуль *'M09-0CB00 - Serial2x'* в слот слева от модуля ЦПУ. По умолчанию, порт X1 настроен на выполнение функционала PtP.



## 4.1.2 Конфигурирование SLIO CPU 013C

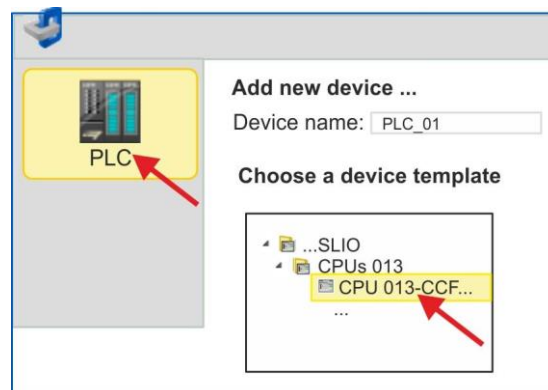
## Добавление модуля ЦПУ в проект

Используйте для конфигурирования *SPEED7 Studio V1.7.1* и выше.

1. ▶ Запустите *SPEED7 Studio*.

## 2. ▶ Создайте новый проект на стартовой странице с помощью команды 'New project' и задайте ему имя с помощью 'Project name'.

⇒ Новый проект создается и отображается в окне 'Devices and networking'.

3. ▶ Кликните в дереве проекта *Project tree* по 'Add new device ...'.

⇒ Откроется диалоговое окно выбора устройства.

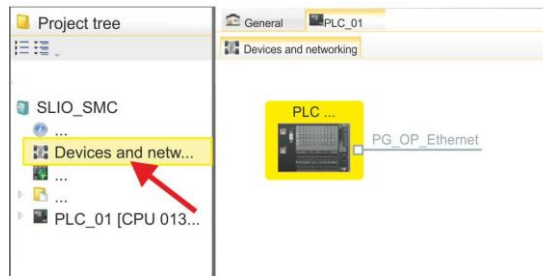
## 4. ▶ Выберите из 'Device templates' процессорный модуль SLIO CPU 013-CCF0R00 и кликните по [OK].

⇒ ЦПУ будет добавлен в раздел 'Devices and networking' и откроется окно 'Device configuration'.



**Конфигурирование порта Ethernet PG/OP**

1. Кликните в дереве проекта *Project tree* по *'Devices and networking'*.  
⇒ Вы получите графическое представление объекта используемого ЦПУ.

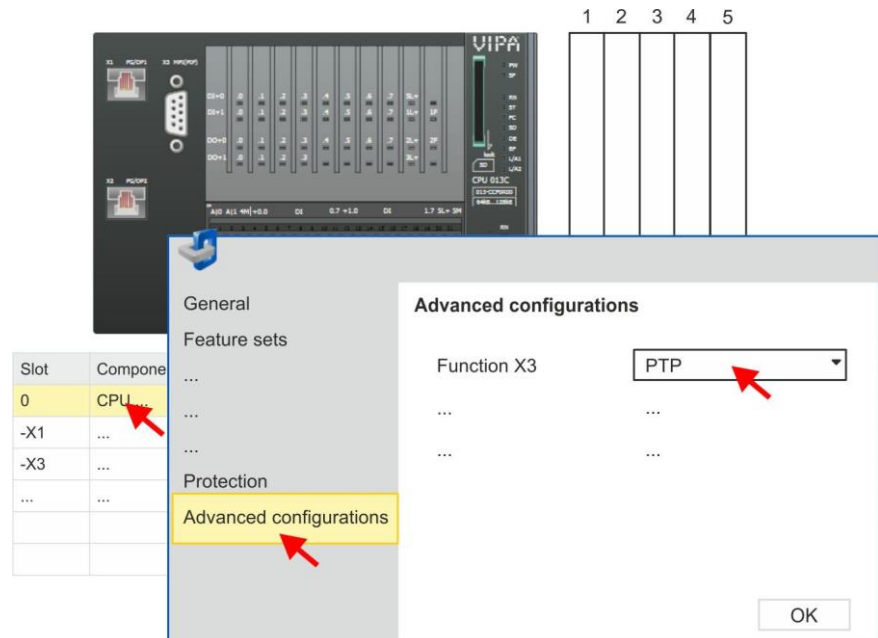


2. Кликните по изображению сети *'PG\_OP\_Ethernet'*.
3. Выберите *'Context menu → Interface properties'*.  
⇒ Откроется диалоговое окно. В нем необходимо ввести IP-адрес для порта Ethernet PG/OP. Предварительно необходимо получить у администратора сети допустимый IP-адрес.
4. Подтвердите нажатием по [OK].  
⇒ Данные IP-адреса сохраняются в проекте и будут отображены в окне *'Local components'* раздела *'Devices and networking'*.

После загрузки проекта в используемый ЦПУ можно получить доступ к устройству через порт Ethernet PG/OP с использованием заданного для него IP-адреса.

**Активирование функциональности PTP**

1. В менеджере проекта *Project tree* кликните по *'PLC... > Device configuration'*.
2. В *'Device configuration'* кликните по *'0 CPU 013...'* и выберите *'Context menu → Components properties'*.  
⇒ Откроется диалоговое окно свойств.



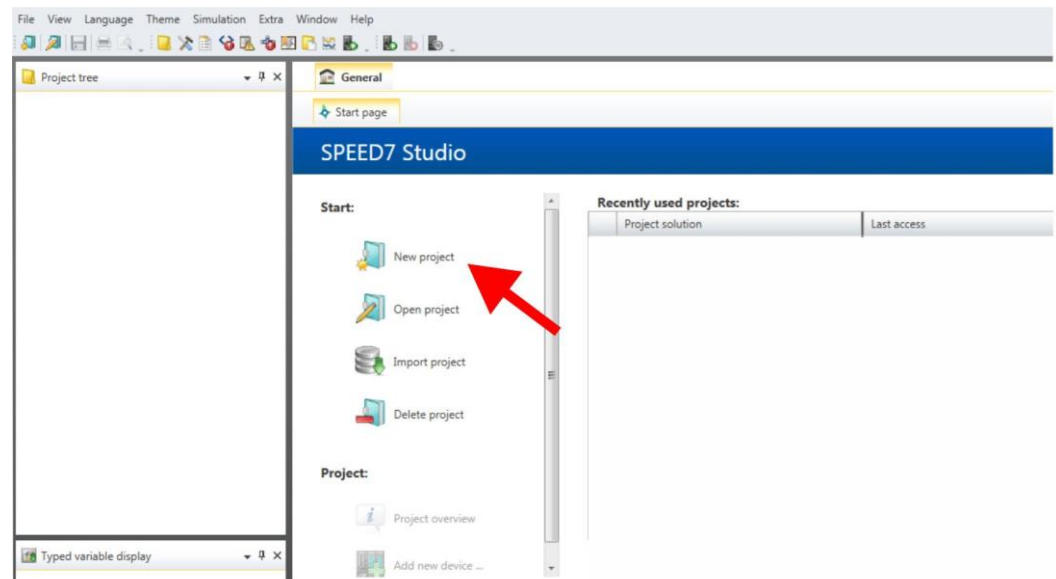
3. Кликните по *'Advanced configurations'* и для *'Function X3'* задайте значение *'PTP'*.

## 4.1.3 Конфигурирование SLIO CPU 014 ... 017

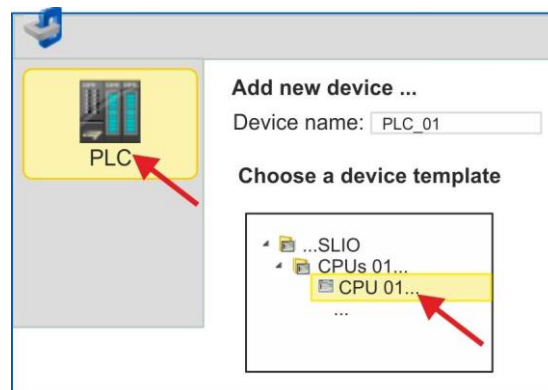
## Добавление модуля ЦПУ в проект

Используйте для конфигурирования *SPEED7 Studio V1.7.1* и выше.

1. ▶ Запустите *SPEED7 Studio*.



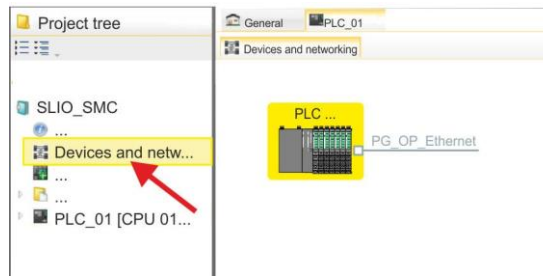
2. ▶ Создайте новый проект на стартовой странице с помощью команды '*New project*' и задайте ему имя с помощью '*Project name*'.  
⇒ Новый проект создается и отображается в окне '*Devices and networking*'.
3. ▶ Кликните в дереве проекта *Project tree* по '*Add new device ...*'.



- ⇒ Откроется диалоговое окно выбора устройства.
4. ▶ Выберите из '*Device templates*' используемый процессорный модуль серии SLIO и кликните по [OK].  
⇒ ЦПУ будет добавлен в разделы '*Devices and networking*' и откроется окно '*Device configuration*'.

**Конфигурирование порта Ethernet PG/OP**

1. Кликните в дереве проекта *Project tree* по *'Devices and networking'*.  
⇒ Вы получите графическое представление используемого ЦПУ.



2. Кликните по изображению сети *'PG\_OP\_Ethernet'*.
3. Выберите *'Context menu → Interface properties'*.  
⇒ Откроется диалоговое окно. В нем необходимо ввести IP-адрес для порта Ethernet PG/OP. Предварительно необходимо получить у администратора сети допустимый IP-адрес.
4. Подтвердите нажатием по [OK].  
⇒ Данные IP-адреса сохраняются в проекте и будут отображены в окне *'Local components'* раздела *'Devices and networking'*.

После загрузки проекта в используемый ЦПУ можно получить доступ к устройству через порт Ethernet PG/OP с использованием заданного для него IP-адреса.

**Активирование функциональности PtP**

В процессорных модулях SLIO CPU 014 ... 017 порт RS-485 стандартно поддерживает обмен в режиме PtP. Для активации функциональности PtP никакое изменение аппаратной конфигурации контроллера не требуется.

**4.2 Прикладная программа**

**4.2.1 Структура программы**

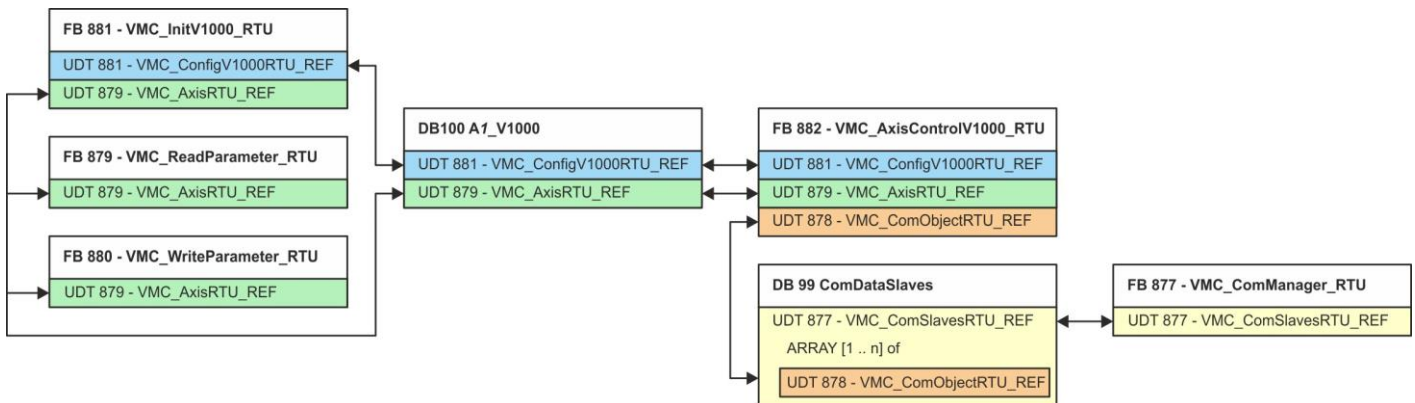
**ОВ 100**

FB 876 - VMC\_ConfigMaster\_RTU ↪ 53

FB 876 - VMC_ConfigMaster_RTU
SFC 216 - SER_CFG

- Этот блок используется для параметрирования последовательного порта процессорного модуля при реализации обмена с использованием протокола Modbus RTU.
- Внутри него вызывается блок SFC 216 - SER\_CFG.

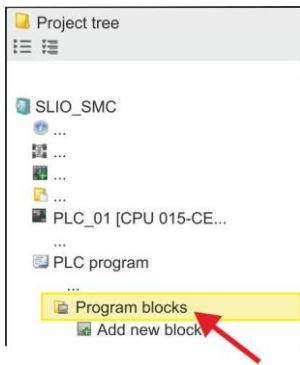
**ОВ 1**



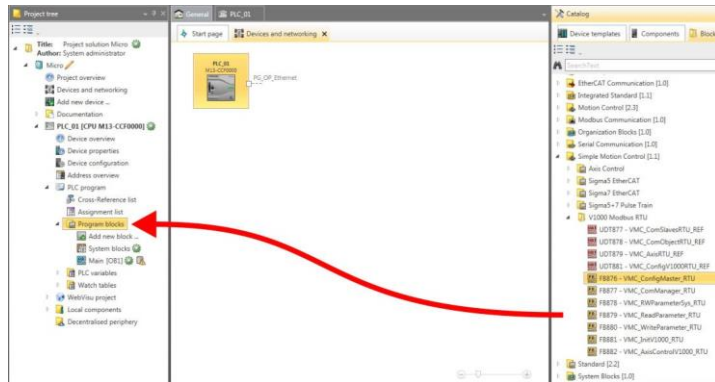
За исключением блоков DB 99 и FB 877, для каждого подключенного частотного преобразователя необходимо создать блоки, перечисленные ниже:

- FB 881 - VMC\_InitV1000\_RTU ↗ 56
  - Блок FB 881 - VMC\_InitV1000\_RTU инициализирует соответствующий преобразователь частоты пользовательскими данными.
  - Прежде чем управлять преобразователем частоты, его необходимо инициализировать.
  - UDT 881 - VMC\_ConfigV1000RTU\_REF ↗ 53
  - UDT 879 - VMC\_AxisRTU\_REF ↗ 53
- FB 879 - VMC\_ReadParameter\_RTU ↗ 55
  - С помощью этого FB обеспечивается чтение параметров из преобразователя частоты через сетевой интерфейс Modbus RTU.
  - Считываемые данные записываются в блок данных.
  - UDT 879 - VMC\_AxisRTU\_REF ↗ 53
- FB 880 - VMC\_WriteParameter\_RTU ↗ 56
  - С помощью этого FB обеспечивается запись параметров в преобразователь частоты через сетевой интерфейс Modbus RTU.
  - Записываемые данные должны храниться в блоке данных.
  - UDT 879 - VMC\_AxisRTU\_REF ↗ 53
- DB 100 - A1\_V1000
  - Для каждого преобразователя частоты, подключенного к сети Modbus RTU, необходимо создать блок данных.
  - UDT 879 - VMC\_AxisRTU\_REF ↗ 53
  - UDT 881 - VMC\_ConfigV1000RTU\_REF ↗ 53
- FB 882 - VMC\_AxisControlV1000\_RTU ↗ 58
  - С помощью этого блока можно через сеть Modbus RTU управлять преобразователем частоты, а также контролировать его состояние.
  - UDT 881 - VMC\_ConfigV1000RTU\_REF ↗ 53
  - UDT 879 - VMC\_AxisRTU\_REF ↗ 53
  - UDT 878 - VMC\_ComObjectRTU\_REF ↗ 53
- DB 99 - ComDataSlaves
  - Для хранения коммуникационных данных преобразователей частоты, подключенных к сети Modbus RTU, должен быть создан общий блок данных.
  - UDT 877 - VMC\_ComSlavesRTU\_REF ↗ 53
  - UDT 878 - VMC\_ComObjectRTU\_REF ↗ 53
- FB 877 - VMC\_ComManager\_RTU ↗ 55
  - Этот блок обеспечивает обмен по сети только с одним преобразователем частоты (ведомым устройством Modbus). Если используется несколько преобразователей частоты, этот блок, выступая в качестве диспетчера связи, отправляет команды в соответствующие ведомые устройства Modbus и оценивает ответы от них.
  - UDT 877 - VMC\_ComSlavesRTU\_REF ↗ 53

#### 4.2.2 Копирование блоков в проект



1. Откройте в менеджере проекта 'Project tree → ...CPU... → PLC program → Program blocks'.



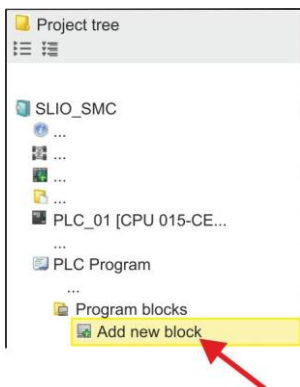
2. Во вкладке 'Catalog' откройте библиотеку 'Simple Motion Control' в разделе 'Blocks' и из папки 'V1000 Modbus RTU' перетащите следующие блоки в раздел 'Program blocks' менеджера проекта Project tree:

- FB 876 - VMC\_ConfigMaster\_RTU
- FB 877 - VMC\_ComManager\_RTU
- FB 878 - VMC\_RWParameterSys\_RTU
- FB 879 - VMC\_ReadParameter\_RTU
- FB 880 - VMC\_WriteParameter\_RTU
- FB 881 - VMC\_InitV1000\_RTU
- FB 882 - VMC\_AxisControlV1000\_RTU

При этом в проект автоматически добавляются следующие блоки:

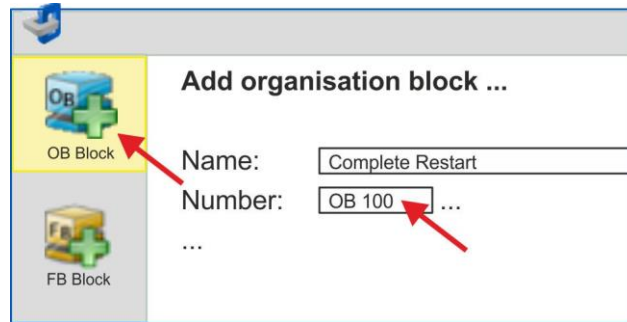
- SEND (FB 60)
- RECEIVE (FB 61)
- RTU\_MB\_MASTER (FB 72)
- SER\_CFG (FC 216)
- SER\_SND (FC 217)
- SER\_RCV (FC 218)
- VMC\_ComSlavesRTU\_REF (UDT 877)
- VMC\_ComObjectRTU\_REF (UDT 878)
- VMC\_ComObjectRTU\_REF (UDT 879)
- VMC\_ComObjectRTU\_REF (UDT 881)

#### 4.2.3 Создание ОБ 100 для последовательного обмена



1. Откройте в менеджере проекта 'Project tree → ...CPU... → PLC program → Program blocks' → Add new block'.

⇒ В ответ откроется диалоговое окно 'Add block'.



2. Введите значение OB 100 и подтвердите нажатием по [OK].  
⇒ OB 100 создается и открывается.
3. Добавьте в OB 100 вызов Call FB876, DB876 .  
⇒ В ответ будет создан вызов блока и откроется диалоговое окно для задания экземплярного блока данных 'VMC\_ConfigMaster\_RTU\_876'.
4. Подтвердите запрос экземплярного блока данных с помощью [OK].
5. Задайте следующие параметры:

Call FB876, DB876 → Раздел 7.5 'FB 876 - VMC\_ConfigMaster\_RTU - Modbus RTU CPU interface' на стр. 53

Baudrate	:= B#16#09	// Скорость передачи: 09 h (9600 бит/с)	IN: BYTE
CharLen	:= B#16#03	// Количество бит данных: 03h (8 бит)	IN: BYTE
Parity	:= B#16#00	// Контроль по четности: 0 (нет)	IN: BYTE
StopBits	:= B#16#01	// Стоповые биты: 1 (1 бит)	IN: BYTE
TimeOut	:= W#16#1FFF	// Время ожидания ответа: 1FFFh (выбрано с большим запасом)	IN: WORD
Valid	:= "ModbusConfigValid"	// Конфигурация	OUT: BOOL
Error	:= "ModbusConfigError"	// Наличие ошибки	OUT: BOOL
ErrorID	:= "ModbusConfigErrorID"	// Дополнительная информация об ошибке	OUT: WORD

**Символьная переменная** Создание символьных переменных осуществляется через 'Context menu → Create / edit symbol'. При этом соответствующие операнды могут быть заданы в диалоговом режиме.

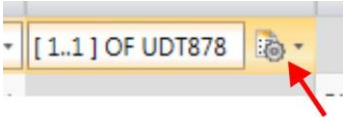
#### 4.2.4 Создание блока данных для ведомого устройства Modbus

Для каждого преобразователя частоты, подключенного к сети Modbus RTU, должен быть создан блок данных.

1. Для этого кликните 'Project tree → ...CPU... → PLC program → Program blocks → Add new block'.  
⇒ В ответ откроется диалоговое окно 'Add block'.
2. Выберите тип блока 'DB block' и задайте для него имя "A1\_V1000". Номер для DB может быть задан любой, например, DB 100. Укажите DB 100 и создайте его как глобальный DB, подтвердив свой выбор нажатием кнопки [OK].  
⇒ Блок создается и открывается.
3. Создайте следующие переменные в "A1\_V1000":
  - 'AxisData' с типом данных UDT 879 - VMC\_AxisRTU\_REF
  - 'V1000Data' с типом данных UDT 881 - VMC\_ConfigV1000RTU\_REF

#### 4.2.5 Задание количества ведомых устройств Modbus

Указание количества частотных преобразователей в сети Modbus RTU осуществляется с помощью UDT 877 - VMC\_ComManager\_RTU.



1. ➤ Откройте UDT 877 - VMC\_ComManager\_RTU.
2. ➤ Для переменной 'Slave' измените тип данных 'Array [1..1] OF', исходя из количества преобразователей частоты в сети Modbus RTU.

Например, при 3 преобразователях частоты, тип данных должен быть изменен на 'Array [1..3] OF'. Чтобы сделать это, кликните по 'Data type settings'.

Обратите внимание, что при этом 'OF UDT 878' остается неизменным.

#### 4.2.6 Создание блока данных для всех ведомых устройств Modbus

Для хранения коммуникационных данных преобразователей частоты, подключенных к сети Modbus RTU, должен быть создан общий блок данных.

1. ➤ Для этого кликните 'Project tree → ...CPU... → PLC program → Program blocks → Add new block'.  
⇒ В ответ откроется диалоговое окно 'Add block'.
2. ➤ Выберите тип блока 'DB block' и задайте ему имя "ComDataSlaves". Номер для DB может быть задан любой, например, DB 99. Укажите DB 99 и создайте его как глобальный DB, подтвердив свой выбор нажатием кнопки [OK].  
⇒ Блок создастся и откроется.
3. ➤ Создайте следующую переменную в "ComDataSlaves":
  - 'Slaves' с типом UDT 877 - VMC\_ComSlavesRTU\_REF.

#### 4.2.7 ОБ 1 - Создание экземплярного DB для диспетчера связи

Блок FB 877 - VMC\_ComManager\_RTU обеспечивает обмен по сети только с одним преобразователем частоты (ведомым устройством Modbus). Выступая в качестве диспетчера связи, этот блок отправляет команды в соответствующие ведомые устройства Modbus и оценивает ответы от них.

1. ➤ Откройте в менеджере проекта 'Project tree → ...CPU... → PLC program → Program blocks → Main [OB1]'.  
⇒ Откроется окно программирования для ОБ 1.
2. ➤ Добавьте в ОБ 1 вызов Call FB877, DB877.  
⇒ В ответ будет создан вызов блока и откроется диалоговое окно для задания экземплярного блока данных 'VMC\_ComManager\_RTU\_877'.
3. ➤ Подтвердите запрос экземплярного блока данных с помощью [OK].
4. ➤ Задайте следующие параметры:

Call FB877, DB877 ↪ Раздел 7.6 'FB 877 - VMC\_ConfigMaster\_RTU - Диспетчер обмена Modbus RTU на стр. 55

NumberOfSlaves	:= 1	// Количество подключенных ПЧ: 1	IN: INT
WaitCycles	:= "ComWaitCycles"	// Минимальное количество циклов ожидания	IN: DINT
SlavesComData	:= "ComDataSlaves.Slave"	// Ссылка на все коммуникационные объекты	IN-OUT: UDT 877

## 4.2.8 ОБ 1 - Создание экземплярного DB для блока инициализации ПЧ V1000

Блок FB 881 - VMC\_InitV1000\_RTU инициализирует соответствующий преобразователь частоты пользовательскими данными. Прежде чем управлять преобразователем частоты, его необходимо инициализировать.

1. ➔ Добавьте в ОБ 1 вызов Call FB881, DB881.  
⇒ В ответ будет создан вызов блока и откроется диалоговое окно для задания экземплярного блока данных 'VMC\_InitV1000\_RTU\_881'.
2. ➔ Подтвердите запрос экземплярного блока данных с помощью [OK].
3. ➔ Задайте следующие параметры:

Call FB881, DB881 ➔ Раздел 7.10 'FB 881 - VMC\_InitV1000\_RTU - Инициализация через Modbus RTU' на стр. 56

Execute	:= "A1_InitExecute"	// Задание запускается по переходу 0-1.	IN: BOOL
Hardware	:= "A1_InitHardware"	// Спецификация используемого оборудования // 1: System SLIO CP040, 2: SPEED7 CPU	IN: BYTE
Laddr	:= "A1_InitLaddr"	// Логический адрес при использовании CP040	IN: INT
UnitId	:= "A1_InitUnitId"	// Адрес Modbus для V1000	IN: BYTE
UserUnitsVelocity	:= "A1_InitUserUnitsVel"	// Пользовательские единицы для скорости: // 0: Гц, 1: %, 2: RPM (об/мин)	IN: INT
UserUnitsAcceleration	:= "A1_InitUserUnitsAcc"	// Пользовательские единицы для // разгона/замедления // 0: 0,01 с, 1: 0,1 с	IN: INT
MaxVelocityApp	:= "A1_InitMaxVelocityApp"	// Макс. скорость в пользовательских единицах	IN: REAL
Done	:= "A1_InitDone"	// Признак завершения задания	OUT: BOOL
Busy	:= "A1_InitBusy"	// Задание в процессе исполнения	OUT: BOOL
Error	:= "A1_InitError"	// Ошибка выполнения	OUT: BOOL
ErrorID	:= "A1_InitErrorID"	// Дополнительная информация об ошибке	OUT: WORD
Axis	:= "A1_V1000".AxisData	// Ссылка на общие данные оси	IN-OUT: UDT 879
V1000	:= "A1_V1000".V1000Data	// Ссылка на индивидуальные данные привода	IN-OUT: UDT 881

**Входные значения**

Все параметры должны быть связаны с соответствующими переменными или операндами. Следующие входные параметры должны быть предварительно назначены:

- Hardware
  - Укажите здесь оборудование, которое будет использоваться для управления преобразователями частоты:
    - 1: Модуль SLIO CP040, логический адрес которого должен быть указан через *Laddr*.
    - 2: Процессорный модуль, выполненный на базе SPEED7.
- Laddr
  - Логический адрес модуля SLIO CP040 (*Hardware* = 1). В противном случае этот параметр игнорируется.
- UnitId
  - Адрес Modbus для V1000.



- UserUnitsVelocity

Пользовательские единицы для скорости:

- 0: Гц

Задается в герцах

- 1: %

Задается в процентах от максимальной скорости =  $2 \cdot f_{\max} / P$

где  $f_{\max}$ : макс. выходная частота (параметр E1-04)

p: Количество полюсов двигателя (индивидуальные параметры двигателя E2-04, E4-04 или E5-04)

- 2: RPM (об/мин)

Данные в оборотах в минуту

- UserUnitsAcceleration

Пользовательские единицы для ускорения и замедления

- 0: 0,01 с (диапазон значений: 0,00 - 600,00 с)

- 1: 0,1 с (диапазон значений: 0,0 - 6000,0 с)

- MaxVelocityApp

Максимальная скорость для приложения. Должна быть задана в пользовательских единицах. Используется в командах перемещения для калибровки.

#### 4.2.9 ОБ 1 - Создание экземплярного DB для блока управления осью V1000

С помощью блока FB 882 - VMC\_AxisControlV1000\_RTU можно управлять преобразователем частоты через сетевой интерфейс Modbus RTU, а также контролировать его состояние.

**1.** ➤ Добавьте в ОБ 1 вызов Call FB882, DB882.

⇒ В ответ будет создан вызов блока и откроется диалоговое окно для задания экземплярного блока данных 'VMC\_AxisControlV1000\_RTU\_882'.

**2.** ➤ Подтвердите запрос экземплярного блока данных с помощью [OK].

**3.** ➤ Задайте следующие параметры:

Call FB882, DB882 ➤ Раздел 7.11 "FB 882 - VMC\_AxisControlV1000\_RTU - Modbus RTU Axis control" на стр. 58

AxisEnabled	:= "A1_AxisEnable"	// Активация оси	IN: BOOL
AxisReset	:= "A1_AxisReset"	// Команда: Сброс ошибки ПЧ V1000.	IN: BOOL
StopExecute	:= "A1_StopExecute"	// Команда: Stop - Останов оси	IN: BOOL
MvVelocityExecute	:= "A1_MvVelocityExecute"	// Команда: MoveVelocity (управление скоростью)	IN: BOOL
Velocity	:= "A1_Velocity"	// Параметр: Значение скорости для MoveVelocity	IN: REAL
AccelerationTime	:= "A1_AccelerationTime"	// Параметр: Время разгона	IN: REAL
DecelerationTime	:= "A1_DecelerationTime"	// Параметр: Время замедления	IN: REAL
JogPositive	:= "A1_JogPositive"	// Команда: JogPos	IN: BOOL
JogNegative	:= "A1_JogNegative"	// Команда: JogNeg	IN: BOOL
JogVelocity	:= "A1_JogVelocity"	// Параметр: Значение скорости для толчкового режима	IN: REAL
JogAccelerationTime	:= "A1_JogAccelerationTime"	// Параметр: Время разгона для толчкового режима	IN: REAL
JogDecelerationTime	:= "A1_JogDecelerationTime"	// Параметр: Время замедления для толчкового режима	IN: REAL
AxisReady	:= "A1_AxisReady"	// Состояние: Готовность оси	OUT: BOOL
AxisEnabled	:= "A1_AxisEnabled"	// Состояние: Активация оси	OUT: BOOL
AxisError	:= "A1_AxisError"	// Состояние: Ошибка оси	OUT: BOOL

AxisErrorID	:= "A1_AxisErrorID"	// Состояние: Дополнительная информация // для <i>AxisError</i>	OUT: WORD
DriveError	:= "A1_DriveError"	// Состояние: Ошибка преобразователя частоты	OUT: BOOL
ActualVelocity	:= "A1_ActualVelocity"	// Состояние: Текущая скорость	OUT: REAL
InVelocity	:= "A1_Velocity"	// Статус заданной скорости	OUT: BOOL
CmdDone	:= "A1_CmdDone"	// Состояние: Команда выполнена	OUT: BOOL
CmdBusy	:= "A1_CmdBusy"	// Состояние: Команда в процессе исполнения	OUT: BOOL
CmdAborted	:= "A1_CmdAborted"	// Состояние: Команда прервана	OUT: BOOL
CmdError	:= "A1_CmdError"	// Состояние: Ошибка команды	OUT: BOOL
CmdErrorID	:= "A1_CmdErrorID"	// Состояние: Дополнительная информация // для <i>CmdError</i>	OUT: WORD
CmdActive	:= "A1_CmdActive"	// Состояние: Активная команда	OUT: INT
DirectionPositive	:= "A1_DirectionPositive"	// Состояние: Направление вращения вперёд	OUT: BOOL
DirectionNegative	:= "A1_DirectionNegative"	// Состояние: Направление вращения назад	OUT: BOOL
Axis	:= "A1_V1000".AxisData	// Ссылка на общие данные оси	IN-OUT: UDT 879
V1000	:= "A1_V1000".V1000Data	// Ссылка на общие данные оси // преобразователя частоты	IN-OUT: UDT 881
AxisComData	:= "ComDataSlaves".Slaves.Slave(1)	// Ссылка на коммуникационные данные	IN-OUT: UDT 878

#### 4.2.10 ОБ 1 - Создание экземплярного DB для блока чтения параметров

С помощью блока FB 879 - VMC\_ReadParameter\_RTU обеспечивается чтение параметров из преобразователя частоты через сетевой интерфейс Modbus RTU. Для хранения значений параметров должен быть создан блок данных.

1. Для этого кликните *'Project tree → ...CPU... → PLC program → Program blocks → Add new block'*.  
⇒ В ответ откроется диалоговое окно *'Add block'*.
2. Выберите тип блока *'DB block'* и задайте ему имя "A1\_TransferData". Номер для DB может быть задан любой, например, DB 98. Укажите DB 98 и создайте его как глобальный DB, подтвердив свой выбор нажатием кнопки [OK].  
⇒ Блок создастся и откроется.
3. Создайте следующие переменные в "A1\_TransferData":
  - *'Data\_0'* с типом WORD
  - *'Data\_1'* с типом WORD
  - *'Data\_2'* с типом WORD
  - *'Data\_3'* с типом WORD
4. Добавьте в ОБ 1 вызов Call FB879, DB879.  
⇒ В ответ будет создан вызов блока и откроется диалоговое окно для задания экземплярного блока данных *'VMC\_ReadParameter\_RTU'*.
5. Подтвердите запрос экземплярного блока данных с помощью [OK].
6. Задайте следующие параметры:

Call FB879, DB879	↪	Раздел 7.8 <i>'FB 879 - VMC_ReadParameter_RTU - Modbus RTU read parameters'</i> на стр. 55
Execute	:= "A1_RdParExecute"	// Задание запускается по переходу 0-1. IN: BOOL
StartAddress	:= "A1_RdParStartAddress"	// Начальный адрес 1-го регистра IN: INT

Quantity	:= "A1_RdParQuantity"	// Количество регистров для чтения	IN: INT
Done	:= "A1_RdParDone"	// Признак завершения задания	IN: REAL
Busy	:= "A1_RdParBusy"	// Задание в процессе исполнения	OUT: BOOL
Error	:= "A1_RdParError"	// Наличие ошибки	OUT: BOOL
ErrorID	:= "A1_RdParErrorID"	// Дополнительная информация об ошибке	OUT: BOOL
Data	:= P#DB98.DBX0.0 BYTES 8	// Место хранения значений параметров	OUT: WORD
Axis	:= "A1_V1000".AxisData	// Ссылка на общие данные оси	IN-OUT: UDT 879



Обратите внимание, что только целые регистры могут быть прочитаны как WORD. Для оценки отдельных бит необходимо поменять местами старшие и младшие байты!

#### 4.2.11 ОБ 1 - Создание экземплярного DB для блока записи параметров

С помощью блока FB 880 - VMC\_ReadParameter\_RTU обеспечивается чтение параметров из преобразователя частоты через сетевой интерфейс Modbus RTU. Для этого блока можно использовать тот же блок данных, что был создан для FB чтения параметров, т.е. DB 98.

1. ➤ Добавьте в ОБ 1 вызов Call FB880, DB880.  
⇒ В ответ будет создан вызов блока и откроется диалоговое окно для задания экземплярного блока данных 'VMC\_WriteParameter\_RTU'.
2. ➤ Подтвердите запрос экземплярного блока данных с помощью [OK].
3. ➤ Задайте следующие параметры:

Call FB880, DB880 ➤ *Раздел 7.9 'FB 880 - VMC\_WriteParameter\_RTU - Modbus RTU write parameters' на стр. 56*

Execute	:= "A1_WrParExecute"	// Задание запускается по переходу 0-1.	IN: BOOL
StartAddre	:= "A1_WrParStartAddress"	// Начальный адрес 1-го регистра	IN: INT
Quantity	:= "A1_WrParQuantity"	// Количество регистров для записи	IN: INT
Done	:= "A1_WrParDone"	// Признак завершения задания	IN: REAL
Busy	:= "A1_WrParBusy"	// Задание в процессе исполнения	OUT: BOOL
Error	:= "A1_WrParError"	// Ошибка выполнения	OUT: BOOL
ErrorID	:= "A1_WrParErrorID"	// Дополнительная информация об ошибке	OUT: BOOL
Data	:= P#DB98.DBX0.0 BYTES 8	// Место хранения значений параметров	OUT: WORD
Axis	:= "A1_V1000".AxisData	// Ссылка на общие данные оси	IN-OUT: UDT 879

## 4.2.12 Последовательность действий

1. ➤ Выполните команду *'Project → Compile all'* и загрузите проект в ЦПУ. Дополнительную информацию о процедуре загрузки можно найти в интерактивной справке по *SPEED7 Studio*.  
⇒ Теперь можно использовать свое приложение через установленное коммуникационное соединение.

**ВНИМАНИЕ!**

Всегда соблюдайте указания по технике безопасности для преобразователя частоты, особенно при вводе его в эксплуатацию!

2. ➤ Таблица контроля переменных позволяет вручную управлять преобразователем частоты. Дважды кликните по *'Project tree → ...CPU... → PLC program → Watch tables → Add watch table'*.
3. ➤ Введите имя для таблицы контроля переменных, например, *'V1000'* и подтвердите выбор, кликнув по [OK].  
⇒ Таблица будет создана и открыта для редактирования.
4. ➤ Сначала задайте время ожидания между двумя заданиями. Для преобразователя частоты V1000 это значение должно быть не менее 200 мс. Для этого в таблице контроля переменных в столбце *'Name'* задайте идентификатор *'ComWaitCycles'* как *'Decimal'* и введите в *'Control value'* значение в диапазоне от 200 до 400.



*Впоследствии для повышения производительности обмена это значение можно будет уменьшить до уровня, при котором еще не будет возникать ошибка по тайм-ауту (80C8h). Обратите внимание, что некоторые команды, такие как MoveVelocity могут состоять из нескольких заданий.*

5. ➤ Прежде начать управлять преобразователем частоты, его необходимо инициализировать с помощью FB 881 - VMC\_InitV1000\_RTU ➤ *Раздел 7.10 'FB 881 - VMC\_InitV1000\_RTU - Инициализация через Modbus RTU'* на стр. 56.

Для этого в таблице контроля переменных в столбце *'Name'* задайте идентификатор *'A1\_InitExecute'* как *'Boolean'* и введите в *'Control value'* значение *'True'*. Активируйте *'Control'* и начните передачу значений управления.

- ⇒ Преобразователь частоты инициализируется. После выполнения выход *Done* примет значение TRUE. При возникновении ошибки ее можно идентифицировать с помощью *ErrorID*.



*Продолжайте работу только в том случае, если блок Init не сообщает об ошибке!*

6. ➤ После успешной инициализации обработка регистров подключенных преобразователей частоты происходит циклически, т.е. они получают запросы циклически. При ручном управлении можно использовать FB 882 - VMC\_AxisControlV1000\_RTU для отправки команд в соответствующий преобразователь частоты. ➤ *Раздел 7.11 'FB 882 - VMC\_AxisControlV1000\_RTU - Modbus RTU Axis control'* на стр. 58.
7. ➤ Для этой цели создайте в таблице контроля переменных соответствующие идентификаторы для FB 882 - VMC\_AxisControlV1000\_RTU.
8. ➤ Активируйте соответствующую ось путем установки *AxisEnable*. После установления значения *AxisReady* = TRUE можно приступить к ее управлению с помощью соответствующих команд перемещения.

## 5 Использование Siemens SIMATIC Manager

### 5.1 Предпосылки

#### Обзор

- Используйте для работы пакет Siemens SIMATIC Manager с версии V5.5 SP2 и выше.
- Для обеспечения реализации обмена через последовательный канал к ЦПУ серии MICRO необходимо подключить коммуникационный модуль расширения. Конфигурирование выполняется в Siemens SIMATIC Manager с помощью виртуального устройства PROFINET IO. Для реализации этого необходимо добавить устройство PROFINET IO в каталог оборудования с помощью соответствующего GSDML-файла.
- Конфигурирование ЦПУ 013C серии SLIO выполняется в Siemens SIMATIC Manager с помощью виртуального устройства PROFINET IO. Для реализации этого необходимо добавить устройство PROFINET IO в каталог оборудования с помощью соответствующего GSDML-файла.
- В процессорных модулях SLIO CPUs 014...017 порт RS-485 стандартно поддерживает обмен в режиме PtP. Конфигурирование выполняется в Siemens SIMATIC Manager с помощью виртуального устройства PROFINET IO. Для реализации этого необходимо добавить устройство PROFINET IO в каталог оборудования с помощью соответствующего GSDML-файла.

#### Установка устройства VIPA IO

Установка устройства PROFINET VIPA IO в каталог оборудования осуществляется в следующей последовательности:

1. ➤ Перейдите в сервисную зону сайта [www.vipa.com](http://www.vipa.com).
2. ➤ Загрузите конфигурационный файл для используемого ЦПУ из раздела *'Config files → PROFINET'*.
3. ➤ Распакуйте этот файл в рабочую папку.
4. ➤ Запустите в SIMATIC Manager конфигуратор оборудования (Hardware Configurator).
5. ➤ Закройте все проекты.
6. ➤ Выберите *'Options → Install new GSD file'*.
7. ➤ Перейдите в рабочую папку и установите требуемый файл GSDML.  
⇒ После установки файла описания соответствующее устройство PROFINET IO может быть найдено в разделе *'PROFINET IO → Additional field devices → I/O → VIPA ...'*.

### 5.2 Конфигурирование аппаратных средств

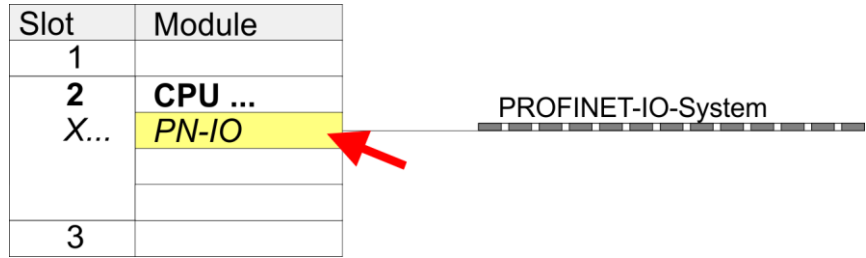
#### 5.2.1 Конфигурирование контроллера серии MICRO

##### Добавление модуля ЦПУ в проект

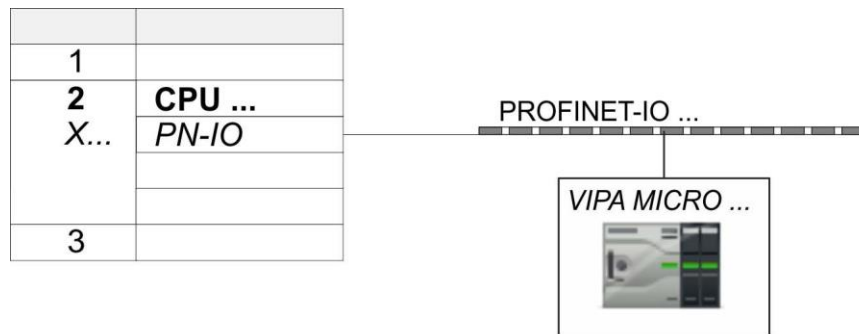
Slot	Module
1	
<b>2</b>	<b>CPU 314C-2PN/DP</b>
X1	MPI/DP
X2	PN-IO
X2...	Port 1
X2...	Port 2
...	...
3	

Для обеспечения совместимости с Siemens SIMATIC Manager необходимо выполнить следующие действия:

1. ➤ Создайте новый проект и запустите в нем конфигуратор оборудования HW Config.
2. ➤ Установите в окно станции монтажную рейку Rail из каталога оборудования.
3. ➤ Установите в слот 2 модуль CPU 314C-2 PN/DP (6ES7314-6EH04-0AB0 V3.3).
4. ➤ Кликните по submodule *'PN-IO'* модуля ЦПУ.
5. ➤ Выберите *'Context menu → Insert PROFINET IO System'*.



6. Введите в соответствующие поля значения IP-адреса и маски сети. Для создания нового подключения через Ethernet кликните по кнопке [New].
7. Кликните по submodule 'PN-IO' модуля ЦПУ и с помощью 'Context menu → Properties' откройте диалоговое окно настройки свойств.
8. Введите на вкладке 'General' имя устройства в поле 'Device name'. Имя устройства должно быть уникальным в рамках подсети Ethernet.



0	<b>VIPA MICRO ...</b>	<b>M13-CCF0000</b>
X2	M13-CCF0000	
1		
2		
3		
...		

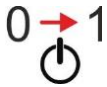
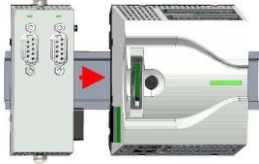
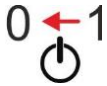
9. В каталоге оборудования перейдите в раздел 'PROFINET IO → Additional field devices → I/O → VIPA ...' и подключите IO-устройство 'M13-CCF0000' к виртуальной шине PROFINET.  
⇒ В окне станции устройства PROFINET IO 'VIPA MICRO PLC' модуль ЦПУ будет помещен в слот 0.

### Конфигурирование порта Ethernet PG/OP

Slot	Module
1	
2	<b>CPU ...</b>
X...	<b>PN-IO</b>
3	
4	<b>343-1EX30</b>
5	
...	

1. Для конфигурирования порта Ethernet PG/OP необходимо установить в слот 4 стойки модуль Siemens CP 343-1 (SIMATIC 300 \ CP 300 \ Industrial Ethernet \ CP 343-1 \ 6GK7 343-1EX30-0XE0 V3.0).
2. Двойным кликом по CP 343-1EX30 откройте диалоговое окно 'Properties' и в нем задайте нужные IP-адрес, маску подсети и адрес шлюза. Предварительно необходимо получить у администратора сети допустимый IP-адрес.
3. Выберите для конфигурируемого CP нужную сеть из списка 'Subnet' или создайте новую, нажав кнопку [New]. Без подключения к подсети данные IP-адреса не устанавливаются!

**Активирование функциональности PtP**



1. ➤ Для активации функциональности PtP никакое изменение аппаратной конфигурации контроллера не требуется. Снимите питание с контроллера.
2. ➤ Смонтируйте коммуникационный модуль расширения.
3. ➤ Установите кабельное подключение с коммуникационным партнером.
4. ➤ Подайте питание на контроллер.  
⇒ Через очень короткое время запуска ЦПУ порт X1 PtP будет готов для обмена данными.

**5.2.2 Конфигурирование SLIO CPU 013C**

**Добавление модуля ЦПУ в проект**

Slot	Module
1	
2	<b>CPU 314C-2PN/DP</b>
X1	MPI/DP
X2	PN-IO
X2...	Port 1
X2...	Port 2
...	...
3	

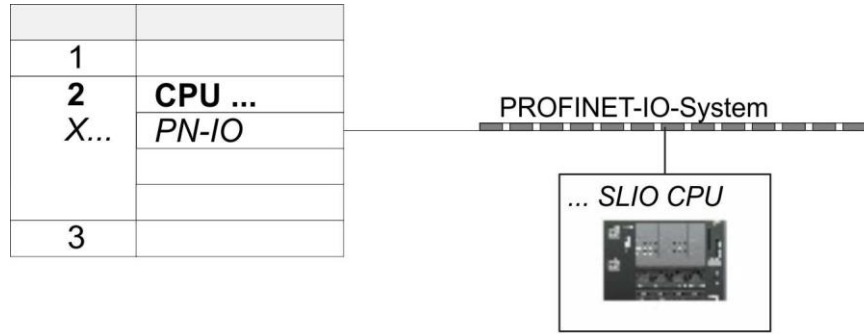
Для обеспечения совместимости с Siemens SIMATIC Manager необходимо выполнить следующие действия:

1. ➤ Создайте новый проект и запустите в нем конфигуратор оборудования HW Config.
2. ➤ Установите в окно станции монтажную рейку Rail из каталога оборудования.
3. ➤ Установите в слот 2 модуль CPU 314C-2 PN/DP (6ES7314-6EH04-0AB0 V3.3).
4. ➤ Кликните по submodule 'PN-IO' модуля ЦПУ.
5. ➤ Выберите 'Context menu → Insert PROFINET IO System'.

Slot	Module
1	
2	<b>CPU ...</b>
X...	<b>PN-IO</b>
3	

PROFINET-IO-System

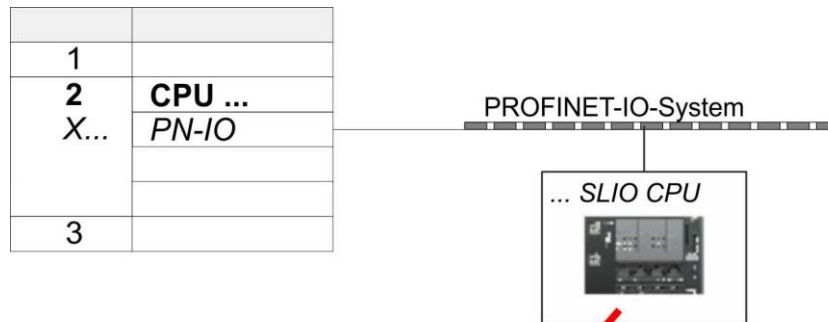
6. ➤ Кликнув по [New], создайте новую подсеть и назначьте подходящий IP-адрес для порта сети PROFINET.
7. ➤ Кликните по submodule 'PN-IO' модуля ЦПУ и с помощью 'Context menu → Properties' откройте диалоговое окно настройки свойств.
8. ➤ Введите на вкладке 'General' имя устройства в поле 'Device name'. Имя устройства должно быть уникальным в рамках подсети Ethernet.



0	... SLIO CPU ...	013-CCF0R00
X2	013-CCF0R00	
1		
2		
3		
...		

9. В каталоге оборудования перейдите в раздел 'PROFINET IO' → 'Additional field devices' → 'I/O' → 'VIPA ...' и подключите IO-устройство '013-CCF0R00' к виртуальной сети PROFINET.  
 ⇒ В окне станции устройства PROFINET IO device 'VIPA MICRO PLC' модуль ЦПУ будет помещён в слот 0. Начиная со слота 1, можно размещать модули расширения системы SLIO.

**Активирование функциональности PTP**



0	... SLIO CPU	013-CCF0R00
X2	013-CCF0R00	
1		
2		
3		
...		

1. Откройте диалоговое окно свойств, дважды кликнув по 'VIPA SLIO CPU'.  
 ⇒ В нем обеспечивается доступ к параметрам, специфичным для ЦПУ VIPA.  
 2. Выберите для 'Function X3' значение 'PTP'.



### Конфигурирование порта Ethernet PG/OP

Slot	Module
1	
2	CPU ...
X...	PN-IO
3	
4	343-1EX30
5	
...	

1. Для конфигурирования порта Ethernet PG/OP необходимо установить в слот 4 стойки модуль Siemens CP 343-1 (SIMATIC 300 \ CP 300 \ Industrial Ethernet \ CP 343-1 \ 6GK7 343-1EX30-0XE0 V3.0).
2. Двойным кликом по CP 343-1EX30 откройте диалоговое окно *'Properties'* и в нем задайте нужные IP-адрес, маску подсети и адрес шлюза. Предварительно необходимо получить у администратора сети допустимый IP-адрес.
3. Выберите для конфигурируемого CP нужную сеть из списка *Subnet'* или создайте новую, нажав кнопку [New]. Без подключения к подсети данные IP-адреса не устанавливаются!

### 5.2.3 Конфигурирование SLIO CPU 014 ... 017

#### Добавление модуля ЦПУ в проект

Slot	Module
1	
2	CPU 315-2 PN/DP
X1	MPI/DP
X2	PN-IO
X2...	Port 1
X2...	Port 2
...	...
3	

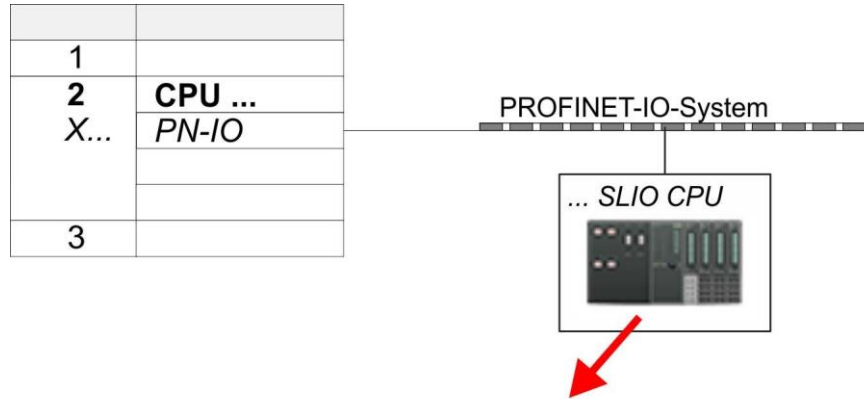
Для обеспечения совместимости с Siemens SIMATIC Manager необходимо выполнить следующие действия:

1. Создайте новый проект и запустите в нем конфигуратор оборудования HW Config.
2. Установите в окно станции монтажную рейку Rail из каталога оборудования.
3. Установите в слот 2 модуль CPU 315C-2 PN/DP (6ES7315-2EH14-0AB0 V3.2).
4. Кликните по submodule *'PN-IO'* модуля ЦПУ.

Slot	Module
1	
2	CPU ...
X...	PN-IO
3	

PROFINET-IO-System

5. Кликнув по [New], создайте новую подсеть и назначьте подходящий IP-адрес для порта сети PROFINET.
6. Кликните по submodule *'PN-IO'* модуля ЦПУ и с помощью *'Context menu' → 'Properties'* откройте диалоговое окно настройки свойств.
7. Введите на вкладке *'General'* имя устройства в поле *'Device name'*. Имя устройства должно быть уникальным в рамках подсети Ethernet.



0	... SLIO CPU ...	...	
X2	...		
1			
2			
3			
...			

8. ➔ В каталоге оборудования перейдите в раздел 'PROFINET IO ➔ Additional field devices ➔ I/O ➔ VIPA ...' и подключите IO-устройство, соответствующее используемому ЦПУ, к виртуальной сети PROFINET.  
 ⇒ В окне станции устройства PROFINET IO device 'VIPA SLIO PLC' модуль ЦПУ будет помещён в слот 0. Начиная со слота 1, можно размещать модули расширения системы SLIO.

**Конфигурирование порта Ethernet PG/OP**

Slot	Module
1	
2	CPU ...
X...	PN-IO
3	
4	343-1EX30
5	
...	

1. ➔ Для конфигурирования порта Ethernet PG/OP необходимо установить в слот 4 стойки модуль Siemens CP 343-1 (SIMATIC 300 \ CP 300 \ Industrial Ethernet \ CP 343-1 \ 6GK7 343-1EX30-0XE0 V3.0).  
 2. ➔ Кликом по CP 343-1EX30 откройте диалоговое окно 'Properties' и в нем задайте нужные IP-адрес, маску подсети и адрес шлюза. Предварительно необходимо получить у администратора сети допустимый IP-адрес.  
 3. ➔ Выберите для конфигурируемого CP нужную сеть из списка 'Subnet' или создайте новую, нажав кнопку [New]. Без подключения к подсети данные IP-адреса не устанавливаются!

**Активирование функциональности PtP**

В процессорных модулях SLIO CPU 014 ... 017 порт RS-485 стандартно поддерживает обмен в режиме PtP. Для активации функциональности PtP никакое изменение аппаратной конфигурации контроллера не требуется.

**5.3 Прикладная программа**

**5.3.1 Структура программы**

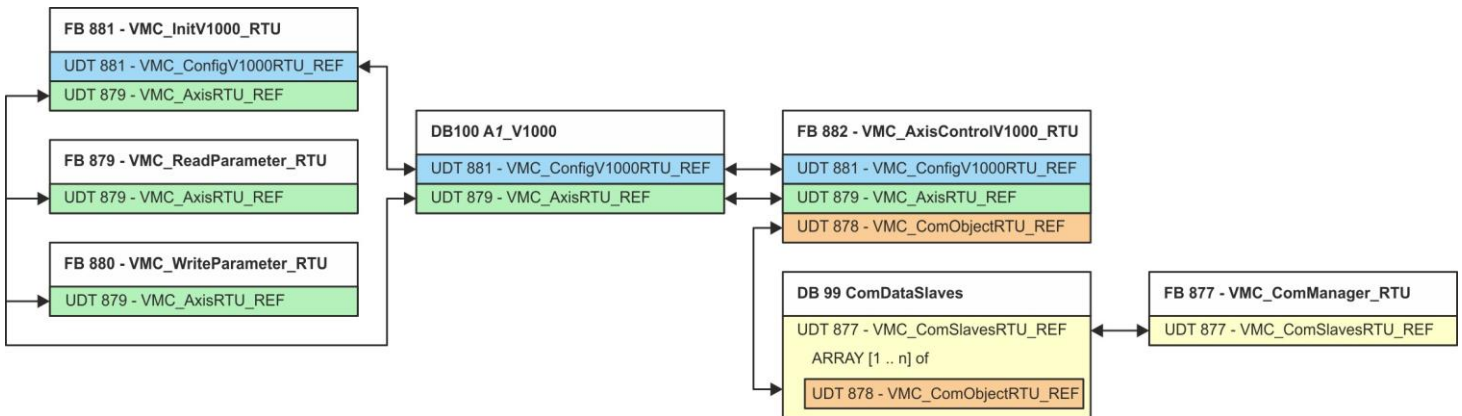
**ОВ 100**

FB 876 - VMC\_ConfigMaster\_RTU ↪ 53

FB 876 - VMC_ConfigMaster_RTU
SFC 216 - SER_CFG

- Этот блок используется для параметрирования последовательного порта процессорного модуля при реализации обмена с использованием протокола Modbus RTU.
- Внутри него вызывается блок SFC 216 - SER\_CFG.

## OB 1



За исключением блоков DB 99 и FB 877 для каждого подключенного преобразователя частоты необходимо создать блоки, перечисленные ниже:

- FB 881 - VMC\_InitV1000\_RTU ↪ 56
  - Блок FB 881 - VMC\_InitV1000\_RTU инициализирует соответствующий преобразователь частоты пользовательскими данными.
  - Прежде чем управлять преобразователем частоты, его необходимо инициализировать.
  - UDT 881 - VMC\_ConfigV1000RTU\_REF ↪ 53
  - UDT 879 - VMC\_AxisRTU\_REF ↪ 53
- FB 879 - VMC\_ReadParameter\_RTU ↪ 55
  - С помощью этого FB обеспечивается чтение параметров из преобразователя частоты через сетевой интерфейс Modbus RTU.
  - Считываемые данные записываются в блок данных.
  - UDT 879 - VMC\_AxisRTU\_REF ↪ 53
- FB 880 - VMC\_WriteParameter\_RTU ↪ 56
  - С помощью этого FB обеспечивается запись параметров в преобразователь частоты через сетевой интерфейс Modbus RTU.
  - Записываемые данные должны храниться в блоке данных.
  - UDT 879 - VMC\_AxisRTU\_REF ↪ 53
- DB 100 - A1\_V1000
  - Для каждого преобразователя частоты, подключенного к сети Modbus RTU, должен быть создан блок данных.
  - UDT 879 - VMC\_AxisRTU\_REF ↪ 53
  - UDT 881 - VMC\_ConfigV1000RTU\_REF ↪ 53
- FB 882 - VMC\_AxisControlV1000\_RTU ↪ 58
  - С помощью этого блока можно управлять преобразователем частоты через сетевой интерфейс Modbus RTU, а также контролировать его состояние.
  - UDT 881 - VMC\_ConfigV1000RTU\_REF ↪ 53
  - UDT 879 - VMC\_AxisRTU\_REF ↪ 53
  - UDT 878 - VMC\_ComObjectRTU\_REF ↪ 53
- DB 99 - ComDataSlaves
  - Для хранения коммуникационных данных преобразователей частоты, подключенных к сети Modbus RTU, должен быть создан общий блок данных.
  - UDT 877 - VMC\_ComSlavesRTU\_REF ↪ 53
  - UDT 878 - VMC\_ComObjectRTU\_REF ↪ 53
- FB 877 - VMC\_ComManager\_RTU ↪ 55
  - Этот блок обеспечивает обмен по сети только с одним преобразователем частоты (ведомым устройством Modbus). Если используется несколько преобразователей частоты, этот блок, выступая в качестве диспетчера связи, отправляет задания в соответствующие ведомые устройства Modbus и оценивает их ответы.
  - UDT 877 - VMC\_ComSlavesRTU\_REF ↪ 53

### 5.3.2 Копирование блоков в проект

#### Добавление библиотеки

1. ➤ Перейдите в сервисную зону сайта [www.vipa.com](http://www.vipa.com).
2. ➤ Загрузите библиотеку *Simple Motion Control* из раздела 'VIPA Lib' области загрузки.
3. ➤ Откройте диалоговое окно для выбора ZIP-файла с помощью '*File* ➔ *Retrieve*'.  
 ➤ Выберите нужный ZIP-файл и нажмите по [Open].
4. ➤ Выберите нужный ZIP-файл и нажмите по [Open].
5. ➤ Укажите папку, в которой будут сохранены блоки, и запустите процесс распаковки, нажав по [OK].

#### Копирование блоков в проект

- Откройте библиотеку после распаковки и перетащите все блоки из 'V1000 Modbus RTU' в раздел 'Blocks' проекта:
  - FB 876 - VMC\_ConfigMaster\_RTU
  - FB 877 - VMC\_ComManager\_RTU
  - FB 878 - VMC\_RWParameterSys\_RTU
  - FB 879 - VMC\_ReadParameter\_RTU
  - FB 880 - VMC\_WriteParameter\_RTU
  - FB 881 - VMC\_InitV1000\_RTU
  - FB 882 - VMC\_AxisControlV1000\_RTU
  - FB 60 - SEND
  - FB 61 - RECEIVE
  - FB 72 - RTU MB\_MASTER
  - FC 216 - SER\_CFG
  - FC 217 - SER\_SND
  - FC 218 - SER\_RCV
  - UDT 877 - VMC\_ComSlavesRTU\_REF
  - UDT 878 - VMC\_ComObjectRTU\_REF
  - UDT 879 - VMC\_AxisRTU\_REF
  - UDT 881 - VMC\_ConfigV1000RTU\_REF
  - SFB 4 - TON

### 5.3.3 Создание OB 100 для последовательного обмена

#### Создание OB прерываний

1. ➤ В проекте нажмите по 'Blocks' и выберите '*Context menu* ➔ *Insert new object* ➔ *Organization block*'.  
 ➔ В ответ откроется диалоговое окно '*Properties Organization block*'.
2. ➤ Добавьте в проект блок OB 100.
3. ➤ Откройте OB 100.
4. ➤ Добавьте в OB 100 вызов `Call FB876, DB876`.  
 ➔ В ответ будет создан вызов блока и откроется диалоговое окно для задания экземплярного блока данных '*VMC\_ConfigMaster\_RTU\_876*'.
5. ➤ Задайте следующие параметры:

`Call FB876, DB876` ➔ Раздел 7.5 '*FB 876 - VMC\_ConfigMaster\_RTU - Modbus RTU CPU interface*' на стр. 53

Baudrate	:= B#16#09	// Скорость передачи: 09 h (9600 бит/с)	IN: BYTE
CharLen	:= B#16#03	// Количество бит данных: 03h (8 бит)	IN: BYTE
Parity	:= B#16#00	// Контроль по четности: 0 (нет)	IN: BYTE
StopBits	:= B#16#01	// Стоповые биты: 1 (1 бит)	IN: BYTE

TimeOut	:= W#16#1FFF	// Время ожидания ответа: 1FFFh (значение с большим запасом)	IN: WORD
Valid	:= "ModbusConfigValid"	// Конфигурация	OUT: BOOL
Error	:= "ModbusConfigError"	// Ошибка выполнения	OUT: BOOL
ErrorID	:= "ModbusConfigErrorID"	// Дополнительная информация об ошибке	OUT: WORD

### Символьная переменная

Создание символьных переменных осуществляется через *'Context menu → Edit symbol'*. При этом соответствующие операнды могут быть заданы в диалоговом режиме.

### 5.3.4 Создание блока данных для ведомого устройства Modbus

Для каждого преобразователя частоты, подключенного к сети Modbus RTU, должен быть создан блок данных.

1. В проекте кликните по *'Blocks'* и выберите *'Context menu → Insert new object → Data block'*.  
⇒ В ответ откроется диалоговое окно *'Add block'*.
2. Задайте следующие параметры:
  - Имя и тип
    - Имя для DB в поле *'Name'* может быть задано любое, например, DB 100. Введите DB 100.
    - Установите для *'Type'* значение *'Shared DB'*.
  - Символьное имя
    - Введите "A1\_TransferData".
 Подтвердите ввод, кликнув по [OK].  
⇒ Блок создан.
3. Двойным кликом откройте DB 100 "A1\_V1000".
4. Создайте следующие переменные в "A1\_V1000":
  - *'AxisData'* с типом данных UDT 879 - VMC\_AxisRTU\_REF
  - *'V1000Data'* с типом данных UDT 881 - VMC\_ConfigV1000RTU\_REF

### 5.3.5 Задание количества ведомых устройств Modbus

Указание количества частотных преобразователей в сети Modbus RTU осуществляется с помощью UDT 877 - VMC\_ComManager\_RTU.

1. Откройте UDT 877 - VMC\_ComManager\_RTU в разделе *'Blocks'*.
2. Для переменной *'Slave'* измените тип данных *'Array [1..1]'* в соответствии с количеством преобразователей частоты в сети Modbus RTU.

UDT 877

Address	Name	Type	...
...	Slave	Array[1..1]	
...		"VMC_ComObjectRTU_REF"	
...		END_STRUCT	

Например, при 3 преобразователях частоты, тип данных должен быть изменен на *'Array [1..3]'*.

Обратите внимание, что остальное остается без изменений.

### 5.3.6 Создание блока данных для всех ведомых устройств Modbus

Для хранения коммуникационных данных преобразователей частоты, подключенных к сети Modbus RTU, должен быть создан общий блок данных.

1. ➤ В проекте кликните по *'Blocks'* и выберите *'Context menu → Insert new object → Data block'*.  
⇒ В ответ откроется диалоговое окно *'Add block'*.
2. ➤ Задайте следующие параметры:
  - Имя и тип
    - Имя для DB в поле *'Name'* может быть задано любое, например, DB 99. Введите DB 99.
    - Установите для *'Type'* значение *'Shared DB'*.
  - Символьное имя
    - Введите "ComDataSlaves".
 Подтвердите ввод, кликнув по [OK].  
⇒ Блок создан.
3. ➤ Двойным кликом откройте DB 99 "ComDataSlaves".
4. ➤ Создайте следующую переменную в "ComDataSlaves":
  - *'Slaves'* с типом UDT 877 - VMC\_ComSlavesRTU\_REF.

### 5.3.7 ОБ 1 - Создание экземплярного DB для диспетчера связи

Блок FB 877 - VMC\_ComManager\_RTU обеспечивает обмен по сети только с одним преобразователем частоты (ведомым устройством Modbus). Выступая в качестве диспетчера связи, этот блок отправляет команды в соответствующие ведомые устройства Modbus и оценивает ответы от них.

1. ➤ Откройте ОБ 1.
2. ➤ Добавьте в ОБ 1 вызов `Call FB877, DB877`.  
⇒ В ответ будет создан вызов блока и откроется диалоговое окно для задания экземплярного блока данных *'VMC\_ComManager\_RTU\_877'*.
3. ➤ Подтвердите запрос экземплярного блока данных с помощью [OK].
4. ➤ Задайте следующие параметры:

`Call FB877, DB877` ➤ Раздел 7.6 *'FB 877 - VMC\_ComManager\_RTU - Диспетчер обмена Modbus RTU'* на стр. 55

<code>NumberOfSlaves</code>	<code>:= 1</code>	// Количество подключенных ПЧ: 1	IN: INT
<code>WaitCycles</code>	<code>:= "ComWaitCycles"</code>	// Минимальное количество циклов ожидания	IN: DINT
<code>SlavesComData</code>	<code>:= "ComDataSlaves.Slave"</code>	// Ссылка на все коммуникационные объекты	IN-OUT: UDT 877

### 5.3.8 ОБ 1 - Создание экземплярного DB для блока инициализации ПЧ V1000

Блок FB 881 - VMC\_InitV1000\_RTU инициализирует соответствующий преобразователь частоты пользовательскими данными. Прежде чем управлять преобразователем частоты, его необходимо инициализировать.

1. ➤ Добавьте в ОБ 1 вызов `Call FB881, DB881`.  
⇒ В ответ будет создан вызов блока и откроется диалоговое окно для задания экземплярного блока данных *'VMC\_InitV1000\_RTU\_881'*.
2. ➤ Подтвердите запрос экземплярного блока данных с помощью [OK].

**3. ➤** Задайте следующие параметры:

Call FB881, DB881	☞ <i>Раздел 7.10 'FB 881 - VMC_InitV1000_RTU - Инициализация через Modbus RTU' на стр. 56</i>		
Execute	:= "A1_InitExecute"	// Задание запускается по переходу 0-1.	IN: BOOL
Hardware	:= "A1_InitHardware"	// Спецификация используемого оборудования // 1: System SLIO CP040, 2: Процессорный // модуль, выполненный на базе SPEED7	IN: BYTE
Laddr	:= "A1_InitLaddr"	// Логический адрес при использовании CP040	IN: INT
UnitId	:= "A1_InitUnitId"	// Адрес Modbus для V1000	IN: BYTE
UserUnitsVelocity	:= "A1_InitUserUnitsVel"	// Пользовательские единицы для скорости: // 0: Гц, 1: %, 2: RPM (об/мин)	IN: INT
UserUnitsAcceleration	:= "A1_InitUserUnitsAcc"	// Пользовательские единицы для // разгона/замедления // 0: 0,01 с, 1: 0,1 с	IN: INT
MaxVelocityApp	:= "A1_InitMaxVelocityApp"	// Макс. скорость в пользовательских единицах	IN: REAL
Done	:= "A1_InitDone"	// Признак завершения задания	OUT: BOOL
Busy	:= "A1_InitBusy"	// Задание в процессе исполнения	OUT: BOOL
Error	:= "A1_InitError"	// Ошибка выполнения	OUT: BOOL
ErrorID	:= "A1_InitErrorID"	// Дополнительная информация об ошибке	OUT: WORD
Axis	:= "A1_V1000".AxisData	// Ссылка на общие данные оси	IN-OUT: UDT 879
V1000	:= "A1_V1000".V1000Data	// Ссылка на индивидуальные данные привода	IN-OUT: UDT 881

**Входные значения**

Все параметры должны быть связаны с соответствующими переменными или операндами. Следующие входные параметры должны быть предварительно назначены:

- Hardware
  - Укажите здесь оборудование, которое будет использоваться для управления преобразователями частоты:
    - 1: Модуль SLIO CP040, логический адрес которого должен быть указан через *Laddr*.
    - 2: Процессорный модуль, выполненный на базе SPEED7.
- Laddr
  - Логический адрес модуля SLIO CP040 (*Hardware* = 1). В противном случае этот параметр игнорируется.
- UnitId
  - Адрес Modbus для V1000.
- UserUnitsVelocity
  - Пользовательские единицы для скорости:
    - 0: Гц  
Задается в герцах
    - 1: %  
Задается в процентах от максимальной скорости =  $2 \cdot f_{\max} / P$   
где  $f_{\max}$ : макс. выходная частота (параметр E1-04)  
p: Количество полюсов двигателя (индивидуальные параметры двигателя E2-04, E4-04 или E5-04)
    - 2: RPM (об/мин)  
Данные в оборотах в минуту

- UserUnitsAcceleration  
Пользовательские единицы для ускорения и замедления
  - 0: 0,01 с (диапазон значений: 0,00 - 600,00 с)
  - 1: 0,1 с (диапазон значений: 0,0 - 6000,0 с)
- MaxVelocityApp  
Максимальная скорость для приложения. Должна быть задана в пользовательских единицах. Используется в командах перемещения для калибровки.

### 5.3.9 ОБ 1 - Создание экземплярного DB для блока управления осью V1000

С помощью блока FB 882 - VMC\_AxisControlV1000\_RTU можно управлять преобразователем частоты через сетевой интерфейс Modbus RTU, а также контролировать его состояние.

1. ➤ Добавьте в ОБ 1 вызов Call FB882, DB882.  
⇒ В ответ будет создан вызов блока и откроется диалоговое окно для задания экземплярного блока данных 'VMC\_AxisControlV1000\_RTU\_882'.
2. ➤ Подтвердите запрос экземплярного блока данных с помощью [OK].
3. ➤ Задайте следующие параметры:

Call FB882, DB882 ↪ *Раздел 7.11 "FB 882 - VMC\_AxisControlV1000\_RTU - Modbus RTU Axis control" на стр. 58*

AxisEnabled	:= "A1_AxisEnable"	// Активация оси	IN: BOOL
AxisReset	:= "A1_AxisReset"	// Команда: Сброс ошибки ПЧ V1000.	IN: BOOL
StopExecute	:= "A1_StopExecute"	// Команда: Stop - Останов оси	IN: BOOL
MvVelocityExecute	:= "A1_MvVelocityExecute"	// Команда: MoveVelocity (управление скоростью)	IN: BOOL
Velocity	:= "A1_Velocity"	// Параметр: Значение скорости для MoveVelocity	IN: REAL
AccelerationTime	:= "A1_AccelerationTime"	// Параметр: Время разгона	IN: REAL
DecelerationTime	:= "A1_DecelerationTime"	// Параметр: Время замедления	IN: REAL
JogPositive	:= "A1_JogPositive"	// Команда: JogPos	IN: BOOL
JogNegative	:= "A1_JogNegative"	// Команда: JogNeg	IN: BOOL
JogVelocity	:= "A1_JogVelocity"	// Параметр: Значение скорости для толчкового режима	IN: REAL
JogAccelerationTime	:= "A1_JogAccelerationTime"	// Параметр: Время разгона для толчкового режима	IN: REAL
JogDecelerationTime	:= "A1_JogDecelerationTime"	// Параметр: Время замедления для толчкового режима	IN: REAL
AxisReady	:= "A1_AxisReady"	// Состояние: Готовность оси	OUT: BOOL
AxisEnabled	:= "A1_AxisEnabled"	// Состояние: Активация оси	OUT: BOOL
AxisError	:= "A1_AxisError"	// Состояние: Ошибка оси	OUT: BOOL
AxisErrorID	:= "A1_AxisErrorID"	// Состояние: Дополнительная информация об ошибке для AxisError	OUT: WORD
DriveError	:= "A1_DriveError"	// Состояние: Ошибка преобразователя частоты	OUT: BOOL
ActualVelocity	:= "A1_ActualVelocity"	// Состояние: Текущая скорость	OUT: REAL
InVelocity	:= "A1_InVelocity"	Статус заданной скорости	OUT: BOOL
CmdDone	:= "A1_CmdDone"	// Состояние: Команда выполнена	OUT: BOOL
CmdBusy	:= "A1_CmdBusy"	// Состояние: Команда в процессе исполнения	OUT: BOOL
CmdAborted	:= "A1_CmdAborted"	// Состояние: Команда прервана	OUT: BOOL
CmdError	:= "A1_CmdError"	// Состояние: Ошибка команды	OUT: BOOL
CmdErrorID	:= "A1_CmdErrorID"	// Состояние: Дополнительная информация об ошибке для CmdError	OUT: WORD



CmdActive	:= "A1_CmdActive"	// Состояние: Активная команда	OUT: INT
DirectionPositive	:= "A1_DirectionPositive"	// Состояние: Направление вращения вперёд	OUT: BOOL
DirectionNegative	:= "A1_DirectionNegative"	// Состояние: Направление вращения назад	OUT: BOOL
Axis	:= "A1_V1000".AxisData	// Ссылка на общие данные оси	IN-OUT: UDT 879
V1000	:= "A1_V1000".V1000Data	// Ссылка на общие данные оси // преобразователя частоты	IN-OUT: UDT 881
AxisComData	:= "ComDataSlaves".Slaves.Slave(1)	// Ссылка на коммуникационные данные	IN-OUT: UDT 878

### 5.3.10 ОБ 1 - Создание экземплярного DB для блока чтения параметров

С помощью блока FB 879 - VMC\_ReadParameter\_RTU обеспечивается чтение параметров из преобразователя частоты через сетевой интерфейс Modbus RTU. Для хранения значений параметров должен быть создан блок данных.

1. В проекте кликните по *'Blocks'* и выберите *'Context menu → Insert new object → Data block'*.  
⇒ В ответ откроется диалоговое окно *'Add block'*.
2. Задайте следующие параметры:
  - Имя и тип
    - Имя для DB в поле *'Name'* может быть задано любое, например, DB 98. Введите DB 98.
    - Установите для *'Type'* значение *'Shared DB'*.
  - Символьное имя
    - Введите "A1\_TransferData".
 Подтвердите ввод, кликнув по [OK].  
⇒ Блок создан.
3. Двойным кликом откройте DB 98 "A1\_TransferData".
4. Создайте в "A1\_TransferData" следующие переменные:
  - *'Data\_0'* с типом WORD
  - *'Data\_1'* с типом WORD
  - *'Data\_2'* с типом WORD
  - *'Data\_3'* с типом WORD
5. Добавьте в ОБ 1 вызов Call FB879, DB879.  
⇒ В ответ будет создан вызов блока и откроется диалоговое окно для задания экземплярного блока данных *'VMC\_ReadParameter\_RTU'*.
6. Подтвердите запрос экземплярного блока данных с помощью [OK].
7. Задайте следующие параметры:

Call FB879, DB879 ↪ *Раздел 7.8 'FB 879 - VMC\_ReadParameter\_RTU - Modbus RTU read parameters'* на стр. 55

Execute	:= "A1_RdParExecute"	// Задание запускается по переходу 0-1.	IN: BOOL
StartAddre	:= "A1_RdParStartAddress"	// Начальный адрес 1-го регистра	IN: INT
Quantity	:= "A1_RdParQuantity"	// Количество регистров для чтения	IN: INT
Done	:= "A1_RdParDone"	// Признак завершения задания	IN: REAL
Busy	:= "A1_RdParBusy"	// Задание в процессе исполнения	OUT: BOOL
Error	:= "A1_RdParError"	// Ошибка выполнения	OUT: BOOL
ErrorID	:= "A1_RdParErrorID"	// Дополнительная информация об ошибке	OUT: BOOL

Data	:= P#DB98.DBX0.0 BYTES 8	// Место хранения значений параметров	OUT: WORD
Axis	:= "A1_V1000".AxisData	// Ссылка на общие данные оси	IN-OUT: UDT 879



Обратите внимание, что только целые регистры могут быть прочитаны как WORD. Для оценки отдельных бит необходимо поменять местами старшие и младшие байты!

### 5.3.11 ОБ 1 - Создание экземплярного DB для блока записи параметров

С помощью блока FB 880 - VMC\_ReadParameter\_RTU обеспечивается чтение параметров из преобразователя частоты через сетевой интерфейс Modbus RTU. Для этого блока можно использовать тот же блок данных, что был создан для FB чтения параметров, т.е. DB 98.

- Добавьте в ОБ 1 вызов Call FB880, DB880.  
⇒ В ответ будет создан вызов блока и откроется диалоговое окно для задания экземплярного блока данных 'VMC\_WriteParameter\_RTU'.
- Подтвердите запрос экземплярного блока данных с помощью [OK].
- Задайте следующие параметры:

Call FB880, DB880 ↪ *Раздел 7.9 'FB 880 - VMC\_WriteParameter\_RTU - Modbus RTU write parameters' на стр. 56*

Execute	:= "A1_WrParExecute"	// Задание запускается по переходу 0-1.	IN: BOOL
StartAddre	:= "A1_WrParStartAddress"	// Начальный адрес 1-го регистра	IN: INT
Quantity	:= "A1_WrParQuantity"	// Количество регистров для записи	IN: INT
Done	:= "A1_WrParDone"	// Признак завершения задания	IN: REAL
Busy	:= "A1_WrParBusy"	// Задание в процессе исполнения	OUT: BOOL
Error	:= "A1_WrParError"	// Ошибка выполнения	OUT: BOOL
ErrorID	:= "A1_WrParErrorID"	// Дополнительная информация об ошибке	OUT: BOOL
Data	:= P#DB98.DBX0.0 BYTES 8	// Место хранения значений параметров	OUT: WORD
Axis	:= "A1_V1000".AxisData	// Ссылка на общие данные оси	IN-OUT: UDT 879

### 5.3.12 Последовательность действий

- Сохраните проект с помощью 'Station → Safe and compile'.
- Загрузите проект в ЦПУ.  
⇒ Теперь можно запустить прикладную программу на исполнение.



#### ВНИМАНИЕ!

Всегда соблюдайте указания по технике безопасности для преобразователя частоты, особенно при вводе его в эксплуатацию!

- Таблица контроля переменных позволяет вручную управлять преобразователем частоты. Для создания таблицы контроля переменных выберите 'PLC → Monitor/Modify variables'.  
⇒ Таблица будет создана и открыта для редактирования.

4. ➔ Сначала задайте время ожидания между двумя заданиями. Для преобразователя частоты V1000 это значение должно быть не менее 200 мс. Для этого в таблице контроля переменных в столбце 'Symbol' задайте идентификатор 'ComWaitCycles' как 'Decimal' и введите в 'Control value' значение в диапазоне от 200 до 400.



*Впоследствии для повышения производительности обмена это значение можно будет уменьшить до уровня, при котором еще не будет возникать ошибка по тайм-ауту (80C8h). Обратите внимание, что некоторые команды, такие как MoveVelocity могут состоять из нескольких заданий.*

5. ➔ Прежде начать управлять преобразователем частоты, его необходимо инициализировать с помощью FB 881 - VMC\_InitV1000\_RTU ➔ *Раздел 7.10 'FB 881 - VMC\_InitV1000\_RTU - Инициализация через Modbus RTU' на стр. 56.*

Для этого в таблице контроля переменных в столбце 'Symbol' задайте идентификатор 'A1\_InitExecute' как 'Boolean' и введите в 'Control value' значение 'True'. Активируйте 'Control' и начните передачу значений управления.

⇒ Преобразователь частоты инициализируется. После выполнения выход Done примет значение TRUE. При возникновении ошибки ее можно идентифицировать с помощью ErrorID.



*Продолжайте работу только в том случае, если блок Init не сообщает об ошибке!*

6. ➔ После успешной инициализации обработка регистров подключенных преобразователей частоты происходит циклически, т.е. они получают запросы циклически. В ручном режиме управления можно использовать FB 882 - VMC\_AxisControlV1000\_RTU для отправки команд в соответствующий преобразователь частоты. ➔ *Раздел 7.11 'FB 882 - VMC\_AxisControlV1000\_RTU - Modbus RTU Axis control' на стр. 58.*
7. ➔ Для этой цели создайте в таблице контроля переменных соответствующие идентификаторы для FB 882 - VMC\_AxisControlV1000\_RTU.
8. ➔ Сохраните таблицу контроля переменных под именем, например, 'V1000'.
9. ➔ Активируйте соответствующую ось путем установки AxisEnable. После установления значения AxisReady = TRUE можно приступать к ее управлению с помощью соответствующих команд перемещения.

## 6 Использование Siemens TIA Portal

### 6.1 Предпосылки

#### Обзор

- Используйте для конфигурирования Siemens TIA Portal V14 и выше.
- Для обеспечения реализации обмена через последовательный канал к ЦПУ серии MICRO необходимо подключить коммуникационный модуль расширения. Конфигурирование выполняется в Siemens TIA Portal с помощью виртуального устройства PROFINET IO. Для реализации этого необходимо добавить устройство PROFINET IO в каталог оборудования с помощью соответствующего GSDML-файла.
- Конфигурирование ЦПУ 013C серии SLIO выполняется в Siemens TIA Portal с помощью виртуального устройства PROFINET IO. Для реализации этого необходимо добавить устройство PROFINET IO в каталог оборудования с помощью соответствующего GSDML-файла.
- В процессорных модулях SLIO CPUs 014 ... 017 порт RS-485 стандартно поддерживает обмен в режиме PtP. Конфигурирование выполняется в Siemens TIA Portal с помощью виртуального устройства PROFINET IO. Для реализации этого необходимо добавить устройство PROFINET IO в каталог оборудования с помощью соответствующего GSDML-файла.

## Установка устройства VIPA IO

Установка устройства PROFINET VIPA IO в каталог оборудования осуществляется в следующей последовательности:

1. ➤ Перейдите в сервисную зону сайта [www.vipa.com](http://www.vipa.com).
2. ➤ Загрузите конфигурационный файл для используемого ЦПУ из раздела 'Config files → PROFINET'.
3. ➤ Распакуйте этот файл в рабочую папку.
4. ➤ Запустите Siemens TIA Portal.
5. ➤ Закройте все проекты.
6. ➤ Перейдите в *Project view*.
7. ➤ Выберите 'Options → Install general station description file (GSD)'.
8. ➤ Перейдите в рабочую папку и установите требуемый файл GSDML.

⇒ После его установки обновится каталог оборудования и Siemens TIA Portal закроется.

После перезапуска Siemens TIA Portal соответствующее устройство PROFINET IO может быть найдено в *Other field devices > PROFINET > IO > VIPA GmbH > ....*



Для того, чтобы компоненты VIPA отображались, необходимо деактивировать "Filter" в каталоге оборудования.

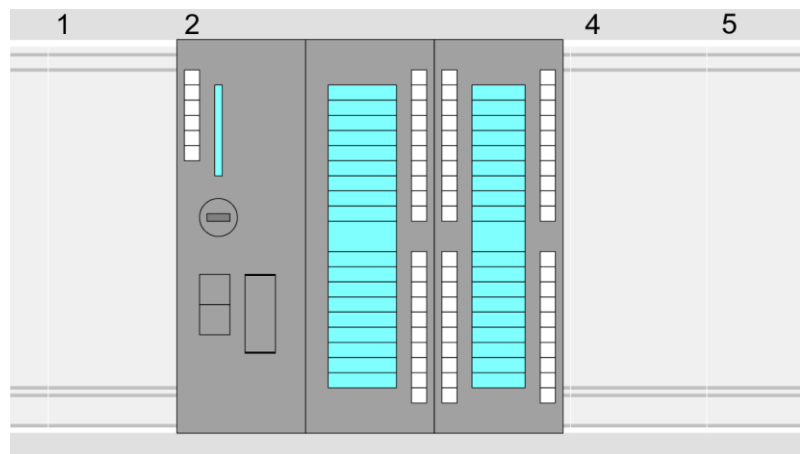
## 6.2 Конфигурирование аппаратных средств

### 6.2.1 Конфигурирование контроллера серии MICRO

#### Добавление модуля ЦПУ в проект

Для обеспечения совместимости с Siemens TIA Portal необходимо выполнить следующие действия:

1. ➤ Запустите Siemens TIA Portal и создайте в нем новый проект.
  2. ➤ Перейдите в *Project view*.
  3. ➤ Кликните в дереве проекта *Project tree* по 'Add new device ...'.
  4. ➤ Выберите следующий модуль ЦПУ в диалоговом окне ввода: SIMATIC S7-300 > CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3)
- ⇒ Модуль ЦПУ вставляется с монтажной рейкой.

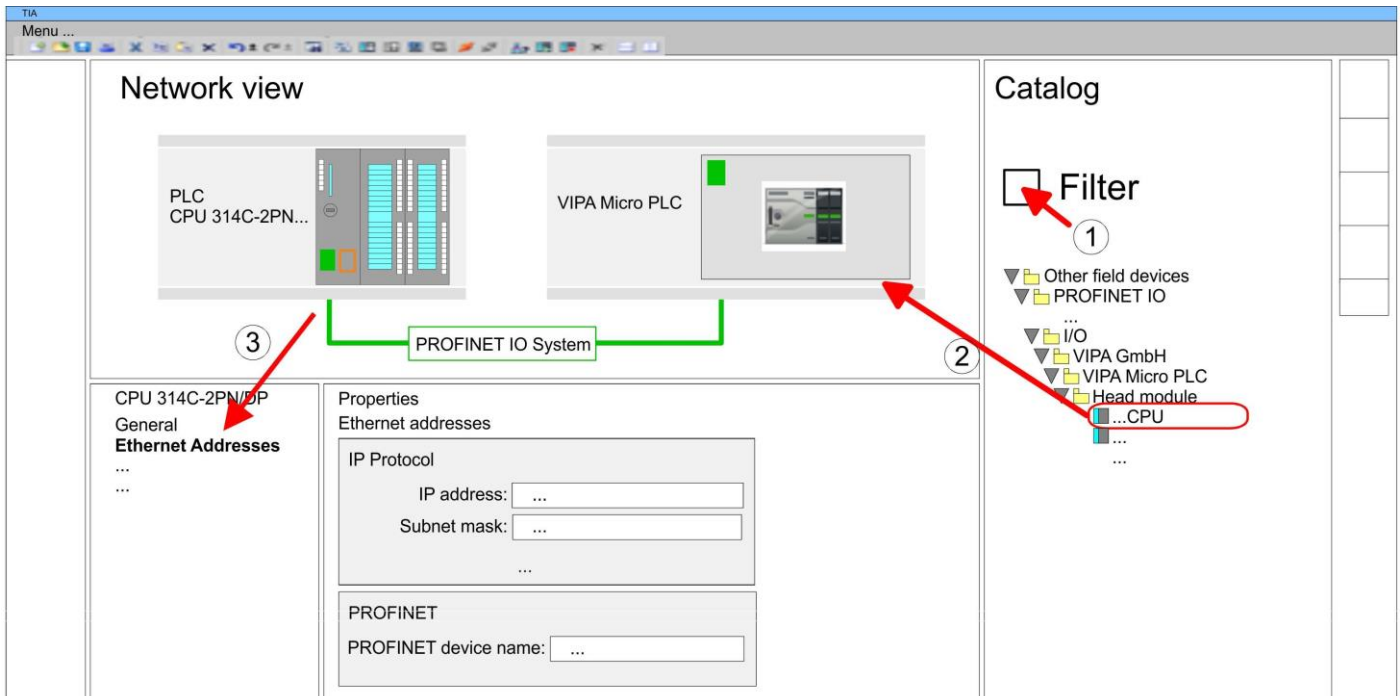


## Device overview

Module	...	Slot	...	Type	...
PLC...		2		CPU 314C-2PN/DP	
MPI interface...		2 X1		MPI/DP interface	
PROFINET interface...		2 X2		PROFINET interface	
DI24/DO16...		2 5		DI24/DO16	
AI5/AO2...		2 6		AI5/AO2	
Count...		2 7		Count	
...					

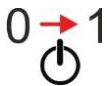
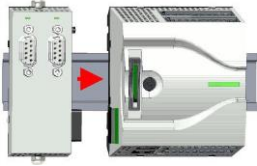
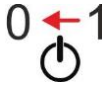
### Подключение ЦПУ как устройство PROFINET IO

1. ➤ Перейдите в *Project area* к *'Network view'*.
2. ➤ После установки файла GSDML устройство ввода/вывода для используемого в примере ЦПУ может быть найдено в каталоге оборудования в разделе *Other field devices > PROFINET > IO > VIPA GmbH > VIPA MICRO PLC*. Свяжите подчиненную систему с ЦПУ, перетащив ее из каталога оборудования в окне *Network view* и подключив ее к ЦПУ через PROFINET.
3. ➤ В окне просмотра *Network view* кликните по порту PROFINET ЦПУ Siemens и затем в окне свойств *'Properties'* введите подходящий IP-адрес в области *'IP protocol'* раздела *'Ethernet address'*.
4. ➤ Введите имя сетевого устройства в поле *'PROFINET device name'* области *'PROFINET'*. Имя устройства должно быть уникальным в пределах подсети Ethernet.



5. ➤ Выберите в окне просмотра *Network view* устройство ввода/вывода *'VIPA MICRO PLC'* и перейдите в *Device overview*.  
 ⇒ В окне станции устройства PROFINET IO *'VIPA MICRO PLC'* модуль ЦПУ будет помещен в слот 0.

**Активирование функциональности PtP**

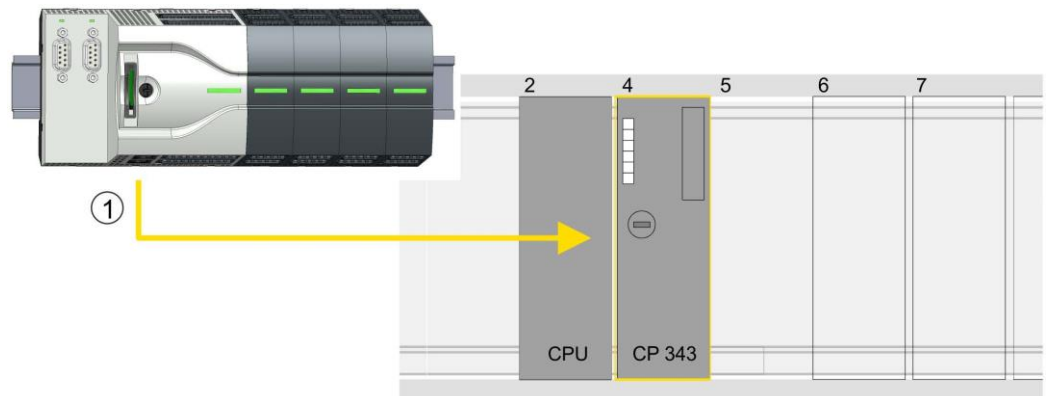


Для активации функциональности PtP никакое изменение аппаратной конфигурации контроллера не требуется.

1. ➤ Снимите питание с контроллера.
2. ➤ Смонтируйте коммуникационный модуль расширения.
3. ➤ Установите кабельное подключение с коммуникационным партнером.
4. ➤ Подайте питание на контроллер.  
⇒ Через очень короткое время после запуска ЦПУ порт X1 PtP будет готов для обмена данными.

**Конфигурирование порта Ethernet PG/OP**

1. ➤ Для конфигурирования порта Ethernet PG/OP поместите в слот 4 стойки модуль Siemens CP 343-1 (6GK7 343-1EX30 0XE0 V3.0).
2. ➤ Двойным кликом по CP 343-1EX30 откройте диалоговое окно 'Properties' и в поле "Ethernet address" введите требуемый IP-адрес. Предварительно необходимо получить у администратора сети допустимый IP-адрес.



1 Порт Ethernet PG/OP

**Device overview**

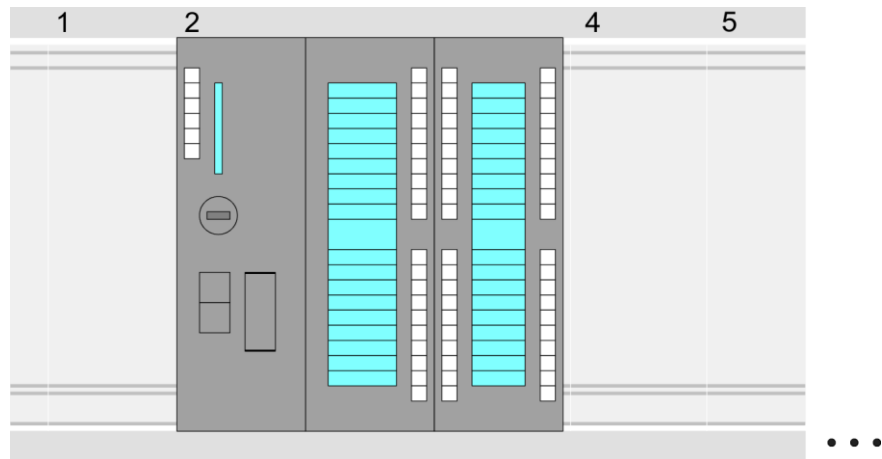
Module	...	Slot	...	Type	...
PLC ...		2		CPU 314C-2PN/DP	
MPI/DP interface		2 X1		MPI/DP interface	
PROFINET interface		2 X2		PROFINET interface	
...		...		...	
CP 343-1		4		CP 343-1	
...		...		...	

## 6.2.2 Конфигурирование SLIO CPU 013C

### Добавление модуля ЦПУ в проект

Для обеспечения совместимости с Siemens TIA Portal необходимо выполнить следующие действия:

1. ➤ Запустите Siemens TIA Portal и создайте в нем новый проект.
2. ➤ Перейдите в *Project view*.
3. ➤ Кликните в дереве проекта *Project tree* по *'Add new device'*.
4. ➤ Выберите следующий модуль ЦПУ в диалоговом окне ввода:  
SIMATIC S7-300 > CPU 314C-2 PN/DP (314-6EH04-0AB0 V3.3)  
⇒ Модуль ЦПУ вставляется с монтажной рейкой.

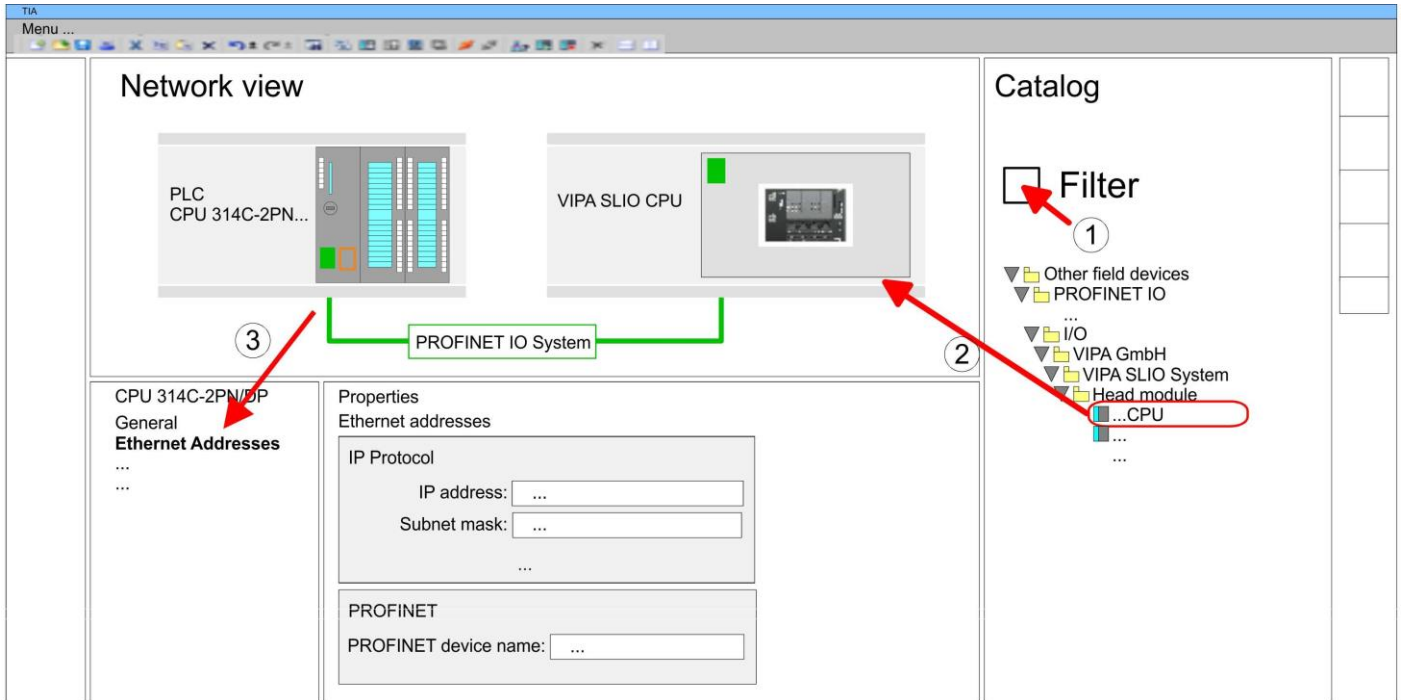


### Device overview

Module	...	Slot	...	Type	...
PLC...		2		CPU 314C-2PN/DP	
MPI interface...		2 X1		MPI/DP interface	
PROFINET interface...		2 X2		PROFINET interface	
DI24/DO16...		2 5		DI24/DO16	
AI5/AO2...		2 6		AI5/AO2	
Count...		2 7		Count	
...					

### Подключение ЦПУ как устройство PROFINET IO

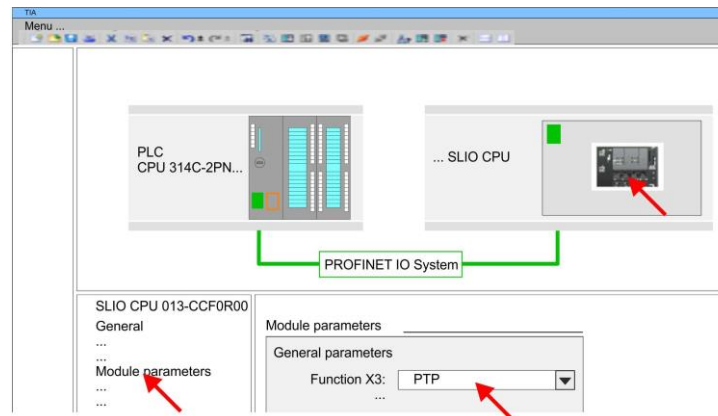
1. ➤ Перейдите в *Project area* к *'Network view'*.
2. ➤ После установки файла GSDML устройство ввода/вывода для SLIO ЦПУ может быть найдено в каталоге оборудования в разделе *Other field devices > PROFINET > IO > VIPA GmbH > VIPA SLIO System*. Свяжите подчиненную систему с ЦПУ, перетащив ее из каталога аппаратного обеспечения в окне *Network view* и подключив ее к ЦПУ через PROFINET.
3. ➤ В окне просмотра *Network view* кликните по порту PROFINET ЦПУ Siemens и затем в окне свойств *'Properties'* введите подходящий IP-адрес в области *'IP protocol'* раздела *'Ethernet address'*.
4. ➤ Введите имя сетевого устройства в поле *'PROFINET device name'* области *'PROFINET'*. Имя устройства должно быть уникальным в рамках подсети Ethernet.



5. Выберите в окне просмотра *Network view* устройство ввода/вывода 'VIPA SLIO CPU' и перейдите в *Device overview*.

⇒ В таблице *Device overview* устройства PROFINET IO 'VIPA SLIO CPU' модуль ЦПУ будет помещён в слот 0.

### Активирование функциональности PтP



1. Откройте диалоговое окно свойств, дважды кликнув по 'VIPA SLIO CPU'.

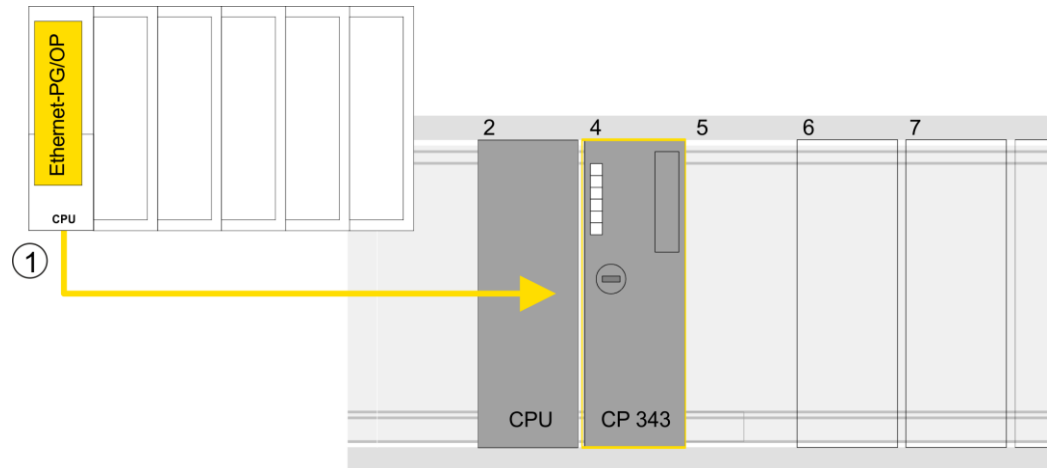
2. Выберите для 'Function X3' значение 'PTP'.

### Конфигурирование порта Ethernet PG/OP

1. Для конфигурирования порта Ethernet PG/OP поместите в слот 4 стойки модуль Siemens CP 343-1 (6GK7 343-1EX30 0XE0 V3.0).

2. Двойным кликом по CP 343-1EX30 откройте диалоговое окно 'Properties' и в поле "Ethernet address" введите требуемый IP-адрес. Предварительно необходимо получить у администратора сети допустимый IP-адрес.





1 Порт Ethernet PG/OP

### Device overview

Module	...	Slot	...	Type	...
PLC ...		2		CPU 315-2 PN/DP	
MPI/DP interface		2 X1		MPI/DP interface	
PROFINET interface		2 X2		PROFINET interface	
...		...		...	
CP 343-1		4		CP 343-1	
...		...		...	

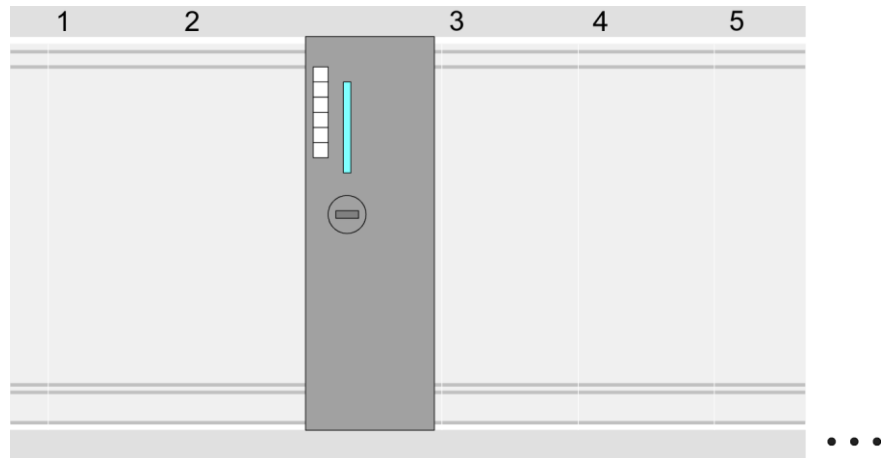
### 6.2.3 Конфигурирование SLIO CPU 014 ... 017

#### Добавление модуля ЦПУ в проект

Для обеспечения совместимости с Siemens TIA Portal необходимо выполнить следующие действия:

1. ➤ Запустите Siemens TIA Portal и создайте в нем новый проект.
2. ➤ Перейдите в *Project view*.
3. ➤ Кликните в дереве проекта *Project tree* по 'Add new device'.

4. ➔ Выберите следующий модуль ЦПУ в диалоговом окне ввода:  
SIMATIC S7-300 > CPU 315-2 PN/DP (315-2EH14-0AB0 V3.2)  
⇒ Модуль ЦПУ вставляется с монтажной рейкой.

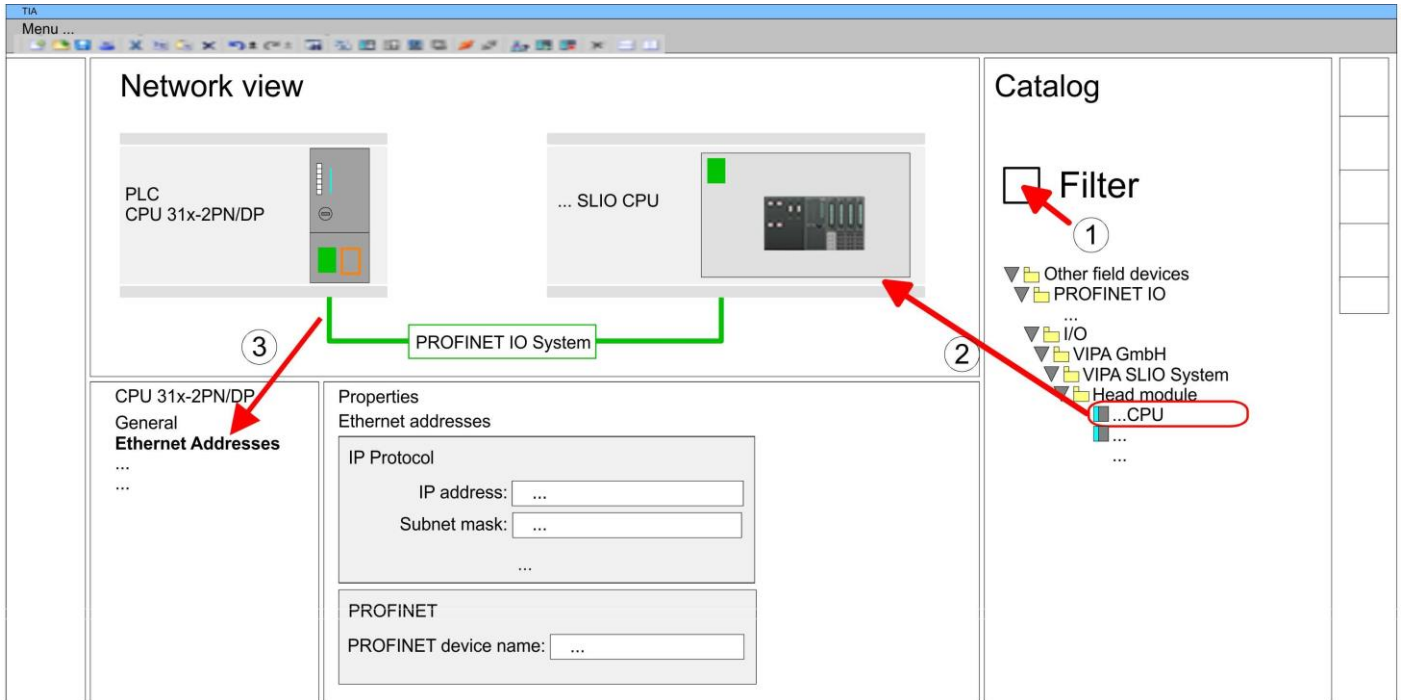


#### Device overview

Module	...	Slot	...	Type	...
PLC ...		2		CPU 315-2 PN/DP	
MPI/DP interface		2 X1		MPI/DP interface	
PROFINET interface		2 X2		PROFINET interface	
...		...		...	

#### Подключение ЦПУ как устройство PROFINET IO

- ➔ Перейдите в *Project area* к *'Network view'*.
- ➔ После установки файла GSDML устройство ввода/вывода для SLIO ЦПУ может быть найдено в каталоге оборудования в разделе *Other field devices > PROFINET > IO > VIPA GmbH > VIPA SLIO System*. Свяжите подчиненную систему с ЦПУ, перетащив ее из каталога аппаратного обеспечения в окне *Network view* и подключив ее к ЦПУ через PROFINET.
- ➔ В окне просмотра *Network view* кликните по порту PROFINET ЦПУ Siemens и затем в окне свойств *'Properties'* введите подходящий IP-адрес в области *'IP protocol'* раздела *'Ethernet address'*.
- ➔ Введите имя сетевого устройства в поле *'PROFINET device name'* области *'PROFINET'*. Имя устройства должно быть уникальным в пределах подсети Ethernet.



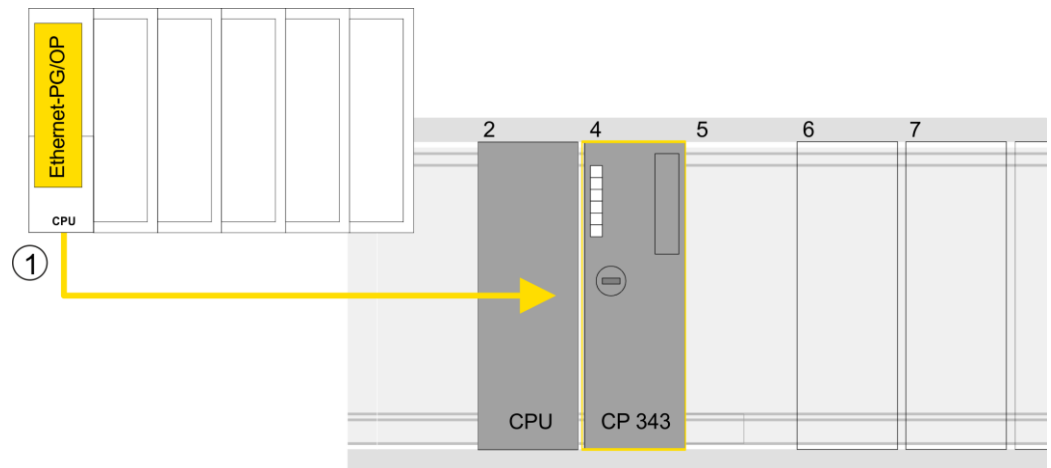
5. Выберите в окне просмотра *Network view* устройство ввода/вывода 'VIPA SLIO CPU' и перейдите в *Device overview*.  
 ⇒ В таблице *Device overview* устройства PROFINET IO 'VIPA SLIO CPU' модуль ЦПУ будет помещён в слот 0.

**Активирование функциональности PtP**

В процессорных модулях SLIO CPU 014 ... 017 порт RS-485 стандартно поддерживает обмен в режиме PtP. Для активации функциональности PtP никакое изменение аппаратной конфигурации контроллера не требуется.

**Конфигурирование порта Ethernet PG/OP**

1. Для конфигурирования порта Ethernet PG/OP поместите в слот 4 стойки модуль Siemens CP 343-1 (6GK7 343-1EX30 0XE0 V3.0).
2. Двойным кликом по CP 343-1EX30 откройте диалоговое окно 'Properties' и в поле "Ethernet address" введите требуемый IP-адрес. Предварительно необходимо получить у администратора сети допустимый IP-адрес.



1 Порт Ethernet PG/OP

**Device overview**

Module	...	Slot	...	Type	...
PLC ...		2		CPU 315-2 PN/DP	
MPI/DP interface		2 X1		MPI/DP interface	
PROFINET interface		2 X2		PROFINET interface	
...		...		...	
CP 343-1		4		CP 343-1	
...		...		...	

**6.3 Прикладная программа**

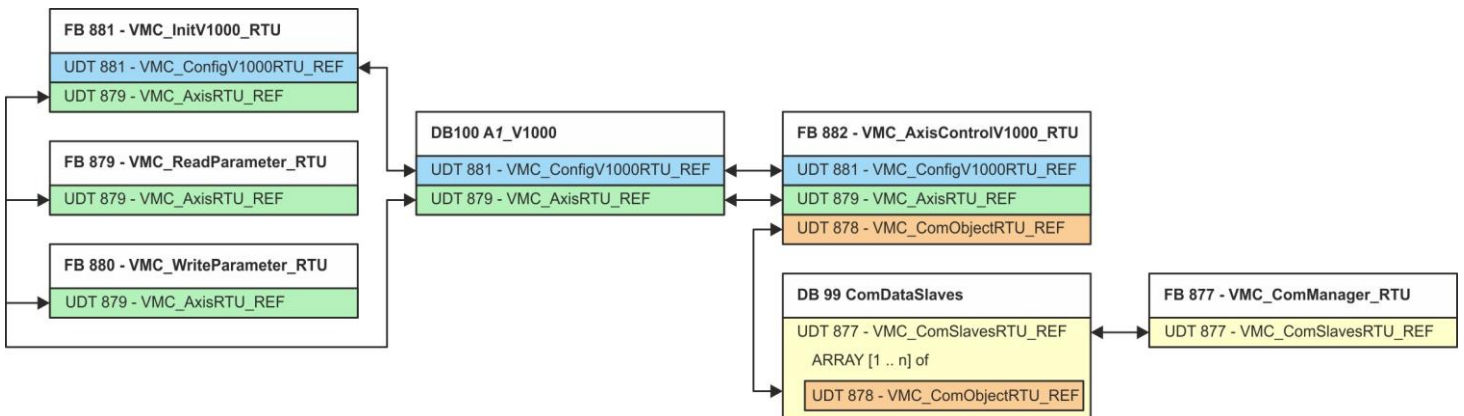
**6.3.1 Структура программы**

**ОВ 100**

FB 876 - VMC_ConfigMaster_RTU
SFC 216 - SER_CFG

- FB 876 - VMC\_ConfigMaster\_RTU ↪ 53
  - Этот блок используется для параметрирования последовательного порта процессорного модуля при реализации обмена с использованием протокола Modbus RTU.
  - Внутри него вызывается блок SFC 216 - SER\_CFG.

**ОВ 1**



За исключением блоков DB 99 и FB 877 для каждого подключенного преобразователя частоты необходимо создать блоки, перечисленные ниже:

- FB 881 - VMC\_InitV1000\_RTU ↪ 56
  - Блок FB 881 - VMC\_InitV1000\_RTU инициализирует соответствующий преобразователь частоты пользовательскими данными.
  - Прежде чем управлять преобразователем частоты, его необходимо инициализировать.
  - UDT 881 - VMC\_ConfigV1000RTU\_REF ↪ 53
  - UDT 879 - VMC\_AxisRTU\_REF ↪ 53
- FB 879 - VMC\_ReadParameter\_RTU ↪ 297
  - С помощью этого FB обеспечивается чтение параметров из преобразователя частоты через сетевой интерфейс Modbus RTU.
  - Считываемые данные записываются в блок данных.
  - UDT 879 - VMC\_AxisRTU\_REF ↪ 53

- FB 880 - VMC\_WriteParameter\_RTU ↗ 298
  - С помощью этого FB обеспечивается чтение параметров из преобразователя частоты через сетевой интерфейс Modbus RTU.
  - Записываемые данные должны храниться в блоке данных.
  - UDT 879 - VMC\_AxisRTU\_REF ↗ 53
- DB 100 - A1\_V1000
  - Для каждого преобразователя частоты, подключенного к сети Modbus RTU, должен быть создан блок данных.
  - UDT 879 - VMC\_AxisRTU\_REF ↗ 53
  - UDT 881 - VMC\_ConfigV1000RTU\_REF ↗ 53
- FB 882 - VMC\_AxisControlV1000\_RTU ↗ 58
  - С помощью этого блока можно управлять преобразователем частоты через сетевой интерфейс Modbus RTU, а также контролировать его состояние.
  - UDT 881 - VMC\_ConfigV1000RTU\_REF ↗ 53
  - UDT 879 - VMC\_AxisRTU\_REF ↗ 53
  - UDT 878 - VMC\_ComObjectRTU\_REF ↗ 53
- DB 99 - ComDataSlaves
  - Для хранения коммуникационных данных преобразователей частоты, подключенных к сети Modbus RTU, должен быть создан общий блок данных.
  - UDT 877 - VMC\_ComSlavesRTU\_REF ↗ 53
  - UDT 878 - VMC\_ComObjectRTU\_REF ↗ 53
- FB 877 - VMC\_ComManager\_RTU ↗ 55
  - Этот блок обеспечивает обмен по сети только с одним преобразователем частоты (ведомым устройством Modbus). Если используется несколько преобразователей частоты, этот блок, выступая в качестве диспетчера связи, отправляет задания в соответствующие ведомые устройства Modbus и оценивает их ответы.
  - UDT 877 - VMC\_ComSlavesRTU\_REF ↗ 53

### 6.3.2 Копирование блоков в проект

#### Добавление библиотеки

1. ➤ Перейдите в сервисную зону сайта [www.vipa.com](http://www.vipa.com).
2. ➤ Загрузите библиотеку *Simple Motion Control* из раздела '*VIPA Lib*' области загрузки. Библиотека доступна в виде zip-файла для соответствующей версии TIA Portal.
3. ➤ Разархивируйте файл ...TIA\_Vxx.zip и скопируйте все полученные файлы и папки в рабочий каталог для Siemens TIA Portal.
4. ➤ Перейдите в *Project view* Siemens TIA Portal.
5. ➤ Выберите вкладку "Libraries" из вертикального меню справа.
6. ➤ Кликните по "Global library".
7. ➤ Кликните по свободной области в '*Global Library*' и выберите '*Context menu* ➔ *Retrieve library*'.
8. ➤ Перейдите в рабочий каталог и загрузите файл ...Simple Motion.zalxx.

**Копирование блоков в проект**

→ Скопируйте все блоки из библиотеки в раздел "Program blocks" менеджера проектов *Project tree* активного проекта.

- FB 876 - VMC\_ConfigMaster\_RTU
- FB 877 - VMC\_ComManager\_RTU
- FB 878 - VMC\_RWParameterSys\_RTU
- FB 879 - VMC\_ReadParameter\_RTU
- FB 880 - VMC\_WriteParameter\_RTU
- FB 881 - VMC\_InitV1000\_RTU
- FB 882 - VMC\_AxisControlV1000\_RTU
- FB 60 - SEND
- FB 61 - RECEIVE
- FB 72 - RTU MB\_MASTER
- FC 216 - SER\_CFG
- FC 217 - SER\_SND
- FC 218 - SER\_RCV
- UDT 877 - VMC\_ComSlavesRTU\_REF
- UDT 878 - VMC\_ComObjectRTU\_REF
- UDT 879 - VMC\_AxisRTU\_REF
- UDT 881 - VMC\_ConfigV1000RTU\_REF
- SFB 4 - TON

**6.3.3 Создание OB 100 для последовательного обмена**

1. → Откройте в менеджере проекта *'Project tree → ...CPU...PLC program → Program blocks → Add new block'*.  
⇒ В ответ откроется диалоговое окно *'Add block'*.
2. → Введите значение OB 100 и подтвердите нажатием по [OK].  
⇒ OB 100 создастся и откроется.
3. → Добавьте в OB 100 вызов `Call FB876, DB876`.  
⇒ В ответ будет создан вызов блока и откроется диалоговое окно для задания экземплярного блока данных *'VMC\_ConfigMaster\_RTU\_876'*.
4. → Подтвердите запрос экземплярного блока данных с помощью [OK].
5. → Задайте следующие параметры:

`Call FB876, DB876` → *Раздел 7.5 'FB 876 - VMC\_ConfigMaster\_RTU - Modbus RTU CPU interface' на стр. 53*

Baudrate	:= B#16#09	// Скорость передачи: 09 h (9600 бит/с)	IN: BYTE
CharLen	:= B#16#03	// Количество бит данных: 03h (8 бит)	IN: BYTE
Parity	:= B#16#00	// Контроль по четности: 0 (нет)	IN: BYTE
StopBits	:= B#16#01	// Стоповые биты: 1 (1 бит)	IN: BYTE
TimeOut	:= W#16#1FFF	// Время ожидания ответа: 1FFFh (значение с // большим запасом)	IN: WORD
Valid	:= "ModbusConfigValid"	// Конфигурация	OUT: BOOL
Error	:= "ModbusConfigError"	// Ошибка выполнения	OUT: BOOL
ErrorID	:= "ModbusConfigErrorID"	// Дополнительная информация об ошибке	OUT: WORD

### 6.3.4 Создание блока данных для ведомого устройства Modbus

Для каждого преобразователя частоты, подключенного к сети Modbus RTU, должен быть создан блок данных.

1. ➤ Откройте в менеджере проекта *'Project tree → ...CPU...PLC program → Program blocks → Add new block'*.  
⇒ В ответ откроется диалоговое окно *'Add block'*.
2. ➤ Выберите тип блока *'DB block'* и задайте для него имя "A1\_V1000". Номер для DB может быть задан любой, например, DB 100. Укажите DB 100 и создайте его как глобальный DB, подтвердив свой выбор нажатием кнопки [OK].  
⇒ Блок создастся и откроется.
3. ➤ Создайте следующие переменные в "A1\_V1000":
  - *'AxisData'* с типом данных UDT 879 - VMC\_AxisRTU\_REF
  - *'V1000Data'* с типом данных UDT 881 - VMC\_ConfigV1000RTU\_REF

### 6.3.5 Задание количества ведомых устройств Modbus

Указание количества частотных преобразователей в сети Modbus RTU осуществляется с помощью UDT 877 - VMC\_ComManager\_RTU.

UDT 877

...	Name	Data type	...
...	Slave	Array[1..1] of ... END_STRUCT	...

1. ➤ Откройте UDT 877 - VMC\_ComManager\_RTU в разделе *'Blocks'*.
2. ➤ Для переменной *'Slave'* измените тип данных *'Array [1..1] OF'* в соответствии с количеством преобразователей частоты в сети Modbus RTU.  
Например, при 3 преобразователях частоты, тип данных должен быть изменен на *'Array [1..3]'*.

Обратите внимание, что остальное остается без изменений.

### 6.3.6 Создание блока данных для всех ведомых устройств Modbus

Для хранения коммуникационных данных преобразователей частоты, подключенных к сети Modbus RTU, должен быть создан общий блок данных.

1. ➤ Откройте в менеджере проекта *'Project tree → ...CPU...PLC program → Program blocks → Add new block'*.  
⇒ В ответ откроется диалоговое окно *'Add block'*.
2. ➤ Выберите тип блока *'DB block'* и задайте ему имя "ComDataSlaves". Номер для DB может быть задан любой, например, DB 99. Укажите DB 99 и создайте его как глобальный DB, подтвердив свой выбор нажатием кнопки [OK].  
⇒ Блок создастся и откроется.
3. ➤ Создайте следующую переменную в "ComDataSlaves":
  - *'Slaves'* с типом UDT 877 - VMC\_ComSlavesRTU\_REF

### 6.3.7 ОБ 1 - Создание экземплярного DB для диспетчера связи

Блок FB 877 - VMC\_ComManager\_RTU обеспечивает обмен по сети только с одним преобразователем частоты (ведомым устройством Modbus). Выступая в качестве диспетчера связи, этот блок отправляет команды в соответствующие ведомые устройства Modbus и оценивает ответы от них.

1. ➤ Откройте ОБ 1.
2. ➤ Добавьте в ОБ 1 вызов Call FB877, DB877.  
⇒ В ответ будет создан вызов блока и откроется диалоговое окно для задания экземплярного блока данных 'VMC\_ComManager\_RTU\_877'.
3. ➤ Подтвердите запрос экземплярного блока данных с помощью [OK].
4. ➤ Задайте следующие параметры:

Call FB877, DB877 ➤ *Раздел 7.6 'FB 877 - VMC\_ComManager\_RTU - Диспетчер обмена Modbus RTU' на стр. 55*

NumberOfSlaves	:= 1	// Количество подключенных ПЧ: 1	IN: INT
WaitCycles	:= "ComWaitCycles"	// Минимальное количество циклов ожидания	IN: DINT
SlavesComData	:= "ComDataSlaves.Slave"	// Ссылка на все коммуникационные объекты	IN-OUT: UDT 877

### 6.3.8 ОБ 1 - Создание экземплярного DB для блока инициализации ПЧ V1000

Блок FB 881 - VMC\_InitV1000\_RTU инициализирует соответствующий преобразователь частоты пользовательскими данными. Прежде чем управлять преобразователем частоты, его необходимо инициализировать.

1. ➤ Добавьте в ОБ 1 вызов Call FB881, DB881.  
⇒ В ответ будет создан вызов блока и откроется диалоговое окно для задания экземплярного блока данных 'VMC\_InitV1000\_RTU\_881'.
2. ➤ Подтвердите запрос экземплярного блока данных с помощью [OK].
3. ➤ Задайте следующие параметры:

Call FB881, DB881 ➤ *Раздел 7.10 'FB 881 - VMC\_InitV1000\_RTU - Инициализация через Modbus RTU' на стр. 56*

Execute	:= "A1_InitExecute"	// Задание запускается по переходу 0-1.	IN: BOOL
Hardware	:= "A1_InitHardware"	// Спецификация используемого оборудования // 1: System SLIO CP040, 2: Процессорный // модуль, выполненный на базе SPEED7	IN: BYTE
Laddr	:= "A1_InitLaddr"	// Логический адрес при использовании CP040	IN: INT
UnitId	:= "A1_InitUnitId"	// Адрес Modbus для V1000	IN: BYTE
UserUnitsVelocity	:= "A1_InitUserUnitsVel"	// Пользовательские единицы для скорости: // 0: Гц, 1: %, 2: RPM (об/мин)	IN: INT
UserUnitsAcceleration	:= "A1_InitUserUnitsAcc"	// Пользовательские единицы для // разгона/замедления // 0: 0,01 с, 1: 0,1 с	IN: INT
MaxVelocityApp	:= "A1_InitMaxVelocityApp"	// Макс. скорость в пользовательских единицах	IN: REAL
Done	:= "A1_InitDone"	// Признак завершения задания	OUT: BOOL
Busy	:= "A1_InitBusy"	// Задание в процессе исполнения	OUT: BOOL
Error	:= "A1_InitError"	// Ошибка выполнения	OUT: BOOL
ErrorID	:= "A1_InitErrorID"	// Дополнительная информация об ошибке	OUT: WORD



Axis	:= "A1_V1000".AxisData	// Ссылка на общие данные оси	IN-OUT: UDT 879
V1000	:= "A1_V1000".V1000Data	// Ссылка на индивидуальные данные привода	IN-OUT: UDT 881

### Входные значения

Все параметры должны быть связаны с соответствующими переменными или операндами. Следующие входные параметры должны быть предварительно назначены:

- **Hardware**  
Укажите здесь оборудование, которое будет использоваться для управления преобразователями частоты:
  - 1: Модуль SLIO CP040, логический адрес которого должен быть указан через *Laddr*.
  - 2: Процессорный модуль, выполненный на базе SPEED7.
- **Laddr**
  - Логический адрес модуля SLIO CP040 (*Hardware* = 1). В противном случае этот параметр игнорируется.
- **UnitId**
  - Адрес Modbus для *V1000*.
- **UserUnitsVelocity**  
Пользовательские единицы для скорости:
  - 0: Гц  
Задается в герцах
  - 1: %  
Задается в процентах от максимальной скорости =  $2 \cdot f_{\max} / P$   
где  $f_{\max}$ : макс. выходная частота (параметр E1-04)  
p: Количество полюсов двигателя (индивидуальные параметры двигателя E2-04, E4-04 или E5-04)
  - 2: RPM (об/мин)  
Данные в оборотах в минуту
- **UserUnitsAcceleration**  
Пользовательские единицы для ускорения и замедления
  - 0: 0,01 с (диапазон значений: 0,00 - 600,00 с)
  - 1: 0,1 с (диапазон значений: 0,0 - 6000,0 с)
- **MaxVelocityApp**  
Максимальная скорость для приложения. Должна быть задана в пользовательских единицах. Используется в командах перемещения для калибровки.

### 6.3.9 ОБ 1 - Создание экземплярного DB для блока управления осью V1000

С помощью блока FB 882 - VMC\_AxisControlV1000\_RTU можно управлять преобразователем частоты через сетевой интерфейс Modbus RTU, а также контролировать его состояние.

1. ➤ Добавьте в ОБ 1 вызов Call FB882, DB882.  
⇒ В ответ будет создан вызов блока и откроется диалоговое окно для задания экземплярного блока данных 'VMC\_AxisControlV1000\_RTU\_882'.
2. ➤ Подтвердите запрос экземплярного блока данных с помощью [OK].
3. ➤ Задайте следующие параметры:

Call FB882, DB882 ↪ Раздел 7.11 "FB 882 - VMC\_AxisControlV1000\_RTU - Modbus RTU Axis control" на стр. 58

AxisEnabled	:= "A1_AxisEnable"	// Активация оси	IN: BOOL
AxisReset	:= "A1_AxisReset"	// Команда: Сброс ошибки ПЧ V1000.	IN: BOOL
StopExecute	:= "A1_StopExecute"	// Команда: Stop - Останов оси	IN: BOOL
MvVelocityExecute	:= "A1_MvVelocityExecute"	// Команда: MoveVelocity (управление // скоростью)	IN: BOOL

Velocity	:= "A1_Velocity"	// Параметр: Значение скорости для MoveVelocity	IN: REAL
AccelerationTime	:= "A1_AccelerationTime"	// Параметр: Время разгона	IN: REAL
DecelerationTime	:= "A1_DecelerationTime"	// Параметр: Время замедления	IN: REAL
JogPositive	:= "A1_JogPositive"	// Команда: <i>JogPos</i>	IN: BOOL
JogNegative	:= "A1_JogNegative"	// Команда: <i>JogNeg</i>	IN: BOOL
JogVelocity	:= "A1_JogVelocity"	// Параметр: Значение скорости для толчкового режима	IN: REAL
JogAccelerationTime	:= "A1_JogAccelerationTime"	// Параметр: Время разгона для толчкового режима	IN: REAL
JogDecelerationTime	:= "A1_JogDecelerationTime"	// Параметр: Время замедления для толчкового режима	IN: REAL
AxisReady	:= "A1_AxisReady"	// Состояние: Готовность оси	OUT: BOOL
AxisEnabled	:= "A1_AxisEnabled"	// Состояние: Активация оси	OUT: BOOL
AxisError	:= "A1_AxisError"	// Состояние: Ошибка оси	OUT: BOOL
AxisErrorID	:= "A1_AxisErrorID"	// Состояние: Дополнительная информация об ошибке для <i>AxisError</i>	OUT: WORD
DriveError	:= "A1_DriveError"	// Состояние: Ошибка преобразователя частоты	OUT: BOOL
ActualVelocity	:= "A1_ActualVelocity"	// Состояние: Текущая скорость	OUT: REAL
InVelocity	:= "A1_Velocity"	// Статус заданной скорости	OUT: BOOL
CmdDone	:= "A1_CmdDone"	// Состояние: Команда выполнена	OUT: BOOL
CmdBusy	:= "A1_CmdBusy"	// Состояние: Команда в процессе исполнения	OUT: BOOL
CmdAborted	:= "A1_CmdAborted"	// Состояние: Команда прервана	OUT: BOOL
CmdError	:= "A1_CmdError"	// Состояние: Ошибка команды	OUT: BOOL
CmdErrorID	:= "A1_CmdErrorID"	// Состояние: Дополнительная информация об ошибке для <i>CmdError</i>	OUT: WORD
CmdActive	:= "A1_CmdActive"	// Состояние: Активная команда	OUT: INT
DirectionPositive	:= "A1_DirectionPositive"	// Состояние: Направление вращения вперёд	OUT: BOOL
DirectionNegative	:= "A1_DirectionNegative"	// Состояние: Направление вращения назад	OUT: BOOL
Axis	:= "A1_V1000".AxisData	// Ссылка на общие данные оси	IN-OUT: UDT 879
V1000	:= "A1_V1000".V1000Data	// Ссылка на общие данные оси преобразователя частоты	IN-OUT: UDT 881
AxisComData	:= "ComDataSlaves".Slaves.Slave(1)	// Ссылка на коммуникационные данные	IN-OUT: UDT 878

### 6.3.10 ОБ 1 - Создание экземплярного DB для блока чтения параметров

С помощью блока FB 879 - VMC\_ReadParameter\_RTU обеспечивается чтение параметров из преобразователя частоты через сетевой интерфейс Modbus RTU. Для хранения значений параметров должен быть создан блок данных.

1. ➤ Откройте в менеджере проекта *'Project tree → ...CPU...PLC program → Program blocks → Add new block'*.  
⇒ В ответ откроется диалоговое окно *'Add block'*.
2. ➤ Выберите тип блока *'DB block'* и задайте ему имя "A1\_TransferData". Номер для DB может быть задан любой. Укажите DB 98 и создайте его как глобальный DB, подтвердив свой выбор нажатием кнопки [OK].  
⇒ Блок создастся и откроется.

3. ➤ Создайте в "A1\_TransferData" следующие переменные:
  - 'Data\_0' с типом WORD
  - 'Data\_1' с типом WORD
  - 'Data\_2' с типом WORD
  - 'Data\_3' с типом WORD
4. ➤ Добавьте в ОБ 1 вызов Call FB879, DB879.  
⇒ В ответ будет создан вызов блока и откроется диалоговое окно для задания экземплярного блока данных 'VMC\_ReadParameter\_RTU'.
5. ➤ Подтвердите запрос экземплярного блока данных с помощью [OK].
6. ➤ Задайте следующие параметры:

Call FB879, DB879 ➤ *Раздел 7.8 'FB 879 - VMC\_ReadParameter\_RTU - Modbus RTU read parameters' на стр. 55*

Execute	:= "A1_RdParExecute"	// Задание запускается по переходу 0-1.	IN: BOOL IN:
StartAddress	:= "A1_RdParStartAddress"	// Начальный адрес 1-го регистра	IN: INT
Quantity	:= "A1_RdParQuantity"	// Количество регистров для чтения	IN: INT
Done	:= "A1_RdParDone"	// Признак завершения задания	IN: REAL
Busy	:= "A1_RdParBusy"	// Задание в процессе исполнения	OUT: BOOL
Error	:= "A1_RdParError"	// Ошибка выполнения	OUT: BOOL
ErrorID	:= "A1_RdParErrorID"	// Дополнительная информация об ошибке	OUT: BOOL
Data	:= P#DB98.DBX0.0 BYTES 8	// Место хранения значений параметров	OUT: WORD
Axis	:= "A1_V1000".AxisData	// Ссылка на общие данные оси	IN-OUT: UDT 879



Обратите внимание, что только целые регистры могут быть прочитаны как WORD. Для оценки отдельных бит необходимо поменять местами старшие и младшие байты!

### 6.3.11 ОБ 1 - Создание экземплярного DB для блока записи параметров

С помощью блока FB 880 - VMC\_WriteParameter\_RTU обеспечивается чтение параметров из преобразователя частоты через сетевой интерфейс Modbus RTU. Для этого блока можно использовать тот же блок данных, что был создан для FB чтения параметров, т.е. DB 98.

1. ➤ Добавьте в ОБ 1 вызов Call FB880, DB880.  
⇒ В ответ будет создан вызов блока и откроется диалоговое окно для задания экземплярного блока данных 'VMC\_WriteParameter\_RTU'.
2. ➤ Подтвердите запрос экземплярного блока данных с помощью [OK].
3. ➤ Задайте следующие параметры:

Call FB880, DB880 ➤ *Раздел 7.9 'FB 880 - VMC\_WriteParameter\_RTU - Modbus RTU write parameters' на стр. 56*

Execute	:= "A1_WrParExecute"	// Задание запускается по переходу 0-1.	IN: BOOL
StartAddress	:= "A1_WrParStartAddress"	// Начальный адрес 1-го регистра	IN: INT
Quantity	:= "A1_WrParQuantity"	// Количество регистров для записи	IN: INT
Done	:= "A1_WrParDone"	// Признак завершения задания	IN: REAL
Busy	:= "A1_WrParBusy"	// Задание в процессе исполнения	OUT: BOOL

Error	:= "A1_WrParError"	// Ошибка выполнения	OUT: BOOL
ErrorID	:= "A1_WrParErrorID"	// Дополнительная информация об ошибке	OUT: BOOL
Data	:= P#DB98.DBX0.0 BYTES 8	// Место хранения значений параметров	OUT: WORD
Axis	:= "A1_V1000".AxisData	// Ссылка на общие данные оси	IN-OUT: UDT 879

### 6.3.12 Последовательность действий

1. ► Сохраните и скомпилируйте проект.
2. ► Загрузите проект в ЦПУ.  
⇒ Теперь можно запустить прикладную программу на исполнение.



#### ВНИМАНИЕ!

Всегда соблюдайте указания по технике безопасности для преобразователя частоты, особенно при вводе его в эксплуатацию!

3. ► Таблица контроля переменных позволяет вручную управлять преобразователем частоты. Для создания таблицы контроля переменных выберите 'Project tree → ...CPU... → Watch and force tables → Add new watch table'.  
⇒ Таблица будет создана и открыта для редактирования.
4. ► Сначала задайте время ожидания между двумя заданиями. Для преобразователя частоты V1000 это значение должно быть не менее 200 мс. Для этого в таблице контроля переменных в столбце 'Name' задайте идентификатор 'ComWaitCycles' как 'DEC' и введите в 'Modify value' значение в диапазоне от 200 до 400.



Впоследствии для повышения производительности обмена это значение можно будет уменьшить до уровня, при котором еще не будет возникать ошибка по тайм-ауту (80С8h). Обратите внимание, что некоторые команды, такие как MoveVelocity могут состоять из нескольких заданий.

5. ► Прежде начать управлять преобразователем частоты, его необходимо инициализировать с помощью FB 881 - VMC\_InitV1000\_RTU. ➔ Раздел 7.10 'FB 881 - VMC\_InitV1000\_RTU - Инициализация через Modbus RTU' на стр. 56

Для этого в таблице контроля переменных в столбце 'Name' задайте идентификатор 'A1\_InitExecute' как 'Boolean' и введите в 'Modify value' значение 'True'. Активируйте режим управления переменными и начните передачу модифицированных значений.

⇒ Преобразователь частоты инициализируется. После выполнения выход Done примет значение TRUE. При возникновении ошибки ее можно идентифицировать с помощью ErrorID.



Продолжайте работу только в том случае, если блок Init не сообщает об ошибке!

6. ► После успешной инициализации обработка регистров подключенных преобразователей частоты происходит циклически, т.е. они получают запросы циклически. В ручном режиме управления можно использовать FB 882 - VMC\_AxisControlV1000\_RTU для отправки команд в соответствующий преобразователь частоты. ➔ Раздел 7.11 'FB 882 - VMC\_AxisControlV1000\_RTU - Modbus RTU Axis control' на стр. 58
7. ► Для этой цели создайте в таблице контроля переменных соответствующие идентификаторы для FB 882 - VMC\_AxisControlV1000\_RTU.
8. ► Сохраните таблицу контроля переменных под именем, например, 'V1000'.

9. ➔ Активируйте соответствующую ось путем установки *AxisEnable*. После установления значения *AxisReady* = TRUE можно приступить к ее управлению с помощью соответствующих команд перемещения.

## 7 Специальные блоки для привода

### 7.1 UDT 877 - VMC\_ComSlavesRTU\_REF - Структура коммуникационных данных Modbus RTU для всех ведомых устройств сети

Это определяемая пользователем структура данных для коммуникационного обмена для всех подключенных ведомых устройств сети Modbus RTU. UDT специально приспособлен для использования с преобразователями частоты при обмене с ними с использованием протокола Modbus RTU.

### 7.2 UDT 878 - VMC\_ComObjectRTU\_REF - Структура коммуникационных данных Modbus RTU для конкретного ведомого устройства сети

Это определяемая пользователем структура коммуникационных данных для конкретного ведомого устройства Modbus RTU. UDT специально приспособлен для использования с преобразователями частоты при обмене с ними с использованием протокола Modbus RTU.

### 7.3 UDT 879 - VMC\_AxisRTU\_REF - Структура данных Modbus RTU для оси

Это определяемая пользователем структура данных, содержащая информацию о состоянии преобразователя частоты. Эта структура служит ссылкой на общие данные оси преобразователя.

### 7.4 UDT 881 - VMC\_ConfigV1000RTU\_REF - Структура данных Modbus RTU для конфигурации

Это определяемая пользователем структура данных, содержащая информацию о конфигурации преобразователя частоты при обмене с ним с использованием протокола Modbus RTU.

### 7.5 FB 876 - VMC\_ConfigMaster\_RTU - Интерфейс Modbus RTU модуля ЦПУ

#### Описание

Этот блок используется для параметрирования последовательного порта процессорного модуля при реализации обмена с использованием протокола Modbus RTU.



*Обратите внимание, что внутри этого блока содержится вызов SFC 216.*

*В SPEED7 Studio этот блок автоматически вставляется в проект.*

*В Siemens SIMATIC Manager блок SFC 216 необходимо скопировать в проект из библиотеки Motion Control Library.*

## Параметр

Параметр	Тип переменной	Тип данных	Описание
Baudrate	ВХОД	BYTE	<p>Скорость передачи данных, бит/с (бод)</p> <ul style="list-style-type: none"> <li>■ 04h: 1200 бит/с</li> <li>■ 05h: 1800 бит/с</li> <li>■ 06h: 2400 бит/с</li> <li>■ 07h: 4800 бит/с</li> <li>■ 08h: 7200 бит/с</li> <li>■ 09h: 9600 бит/с</li> <li>■ 0Ah: 14400 бит/с</li> <li>■ 0Bh: 19200 бит/с</li> <li>■ 0Ch: 38400 бит/с</li> <li>■ 0Dh: 57600 бит/с</li> <li>■ 0Eh: 115200 бит/с</li> </ul>
CharLen	ВХОД	BYTE	<p>Количество бит данных, соответствующих передаваемому символу</p> <ul style="list-style-type: none"> <li>■ 0: 5 бит</li> <li>■ 1: 6 бит</li> <li>■ 2: 7 бит</li> <li>■ 3: 8 бит</li> </ul>
Parity	ВХОД	BYTE	<p>Контроль по чётности. Для реализации контроля по чётности информационные биты дополняются битом чётности со значение "0" или "1" так, чтобы сумма "1" в слове, включая бит чётности, была чётной (при контроле по чётности) или нечётной (при контроле по нечётности). Если контроль по чётности не производится, то бит чётности имеет значение "1", которое не контролируется.</p> <ul style="list-style-type: none"> <li>■ 0: Отсутствует</li> <li>■ 1: Контроль по нечётности</li> <li>■ 2: Контроль по чётности</li> </ul>
StopBits	ВХОД	BYTE	<p>Стоп-биты добавляются к каждому передаваемому символу и отмечают конец символа.</p> <ul style="list-style-type: none"> <li>■ 1: 1 бит</li> <li>■ 2: 1,5 бита</li> <li>■ 3: 2 бита</li> </ul>
TimeOut	ВХОД	WORD (16-разрядное значение)	<p>Время ожидания ответа от ведомого устройства, по истечении которого формируется ошибка обмена.</p> <p>Значение для <i>TimeOut</i> должно быть задано в шестнадцатеричном формате. Шестнадцатеричное значение получается путем умножения желаемого времени в секундах на скорость передачи.</p> <p>Пример: Желаемое время 8 мс при скорости передачи в 19200 бит/с Расчёт:  <math>19200 \text{ бит/с} \times 0,008 \text{ с} \approx 154 \text{ бит} \gggg (9Ah)</math></p> <p>Шестнадцатеричное значение равно 9Ah.</p>
Valid	ВЫХОД	BOOL (Двоичное значение)	<p>Конфигурация</p> <ul style="list-style-type: none"> <li>■ TRUE: Конфигурация действительна.</li> <li>■ FALSE: Конфигурация недействительна.</li> </ul>
Error	ВЫХОД	BOOL (Двоичное значение)	<p>Наличие ошибки</p> <ul style="list-style-type: none"> <li>■ TRUE: Возникла ошибка - см. <i>ErrorID</i>.</li> <li>■ FALSE: Ошибка отсутствует.</li> </ul>
ErrorID	ВЫХОД	WORD (16-разрядное значение)	<p>Дополнительная информация об ошибке</p> <p>↪ <i>Раздел 8 'ErrorID - Дополнительная информация об ошибке на стр. 61</i></p>

## 7.6 FB 877 - VMC\_ComManager\_RTU - Диспетчер обмена Modbus RTU

### Описание

Этот блок упорядочивает коммуникационный обмен так, чтобы только одно ведомое устройство может последовательно осуществлять передачу через последовательный интерфейс. С помощью UDT 877 этот блок имеет доступ к коммуникационным данным всех ведомых устройств.



*В проекте может быть использован только один FB 877 для каждого последовательного интерфейса!*

### Параметр

Параметр	Тип переменной	Тип данных	Описание
NumberOfSlaves	ВХОД	INT	Количество используемых в данный момент ведомых устройств Modbus
WaitCycles	ВХОД	DINT (Двойное целое)	Минимальное число циклов ожидания между двумя запросами ведомого. Это позволяет предотвратить перегрузку ведомого и избежать появления таймаутов.
SlavesComData	ВХОД-ВЫХОД	UDT 877	Ссылка на блок данных со всеми коммуникационными объектами

## 7.7 FB 878 - VMC\_RWParameterSys\_RTU - Чтение/запись параметров системы через Modbus RTU

### Описание

Этот блок используется внутри системы для передачи параметров.



*Нельзя вызывать этот блок, т.к. это может быть привести к сбою в работе системы!*

## 7.8 FB 879 - VMC\_ReadParameter\_RTU - Чтение параметров через Modbus RTU

### Описание

Этот блок может быть использован для чтения параметров из соответствующего ведомого устройства.



*Обратите внимание, что только целые регистры могут быть записаны как WORD. Для установки или сброса отдельных бит необходимо поменять местами старшие и младшие байты!*

### Параметр

Параметр	Тип переменной	Тип данных	Описание
Execute	ВХОД	BOOL (Двоичное значение)	Задание запускается по переходу 0-1.
StartAddress	ВХОД	WORD (16-разрядное значение)	Начальный адрес регистра для выполнения чтения.
Quantity	ВХОД	BYTE	Количество регистров для чтения
Done	ВЫХОД	BOOL (Двоичное значение)	Status ■ TRUE: Команда успешно выполнена
Busy	ВЫХОД	BOOL (Двоичное значение)	Status ■ TRUE: Команда выполняется

Параметр	Тип переменной	Тип данных	Описание
Error	ВЫХОД	BOOL (Двоичное значение)	Status <ul style="list-style-type: none"> <li>TRUE: Возникла ошибка. Дополнительная информация об ошибке может быть взята из параметра <i>ErrorID</i>.</li> </ul>
ErrorID	ВЫХОД	WORD (16-разрядное значение)	Дополнительная информация об ошибке ↗ <i>Раздел 8 'ErrorID - Дополнительная информация об ошибке на стр. 61</i>
Data	ВХОД-ВЫХОД	ANY (любой)	Ссылка на место, где хранить считанные данные
Axis	ВХОД-ВЫХОД	UDT 879	Ссылка на общие данные оси преобразователя частоты

## 7.9 FB 880 - VMC\_WriteParameter\_RTU - Запись параметров через Modbus RTU

### Описание

Этот блок может быть использован для записи параметров в регистры соответствующего ведомого устройства.



*Обратите внимание, что только целые регистры могут быть записаны как WORD. Для установки или сброса отдельных бит необходимо поменять местами старшие и младшие байты!*

### Параметр

Параметр	Тип переменной	Тип данных	Описание
Execute	ВХОД	BOOL (Двоичное значение)	Задание запускается по переходу 0-1.
StartAddress	ВХОД	WORD (16-разрядное значение)	Начальный адрес регистра для выполнения записи.
Quantity	ВХОД	BYTE	Количество регистров для записи
Done	ВЫХОД	BOOL (Двоичное значение)	Status <ul style="list-style-type: none"> <li>TRUE: Задание успешно выполнено</li> </ul>
Busy	ВЫХОД	BOOL (Двоичное значение)	Status <ul style="list-style-type: none"> <li>TRUE: Задание выполняется</li> </ul>
Error	ВЫХОД	BOOL (Двоичное значение)	Status <ul style="list-style-type: none"> <li>TRUE: Возникла ошибка. Дополнительная информация об ошибке может быть взята из параметра <i>ErrorID</i>.</li> </ul>
ErrorID	ВЫХОД	WORD (16-разрядное значение)	Дополнительная информация об ошибке ↗ <i>Раздел 8 'ErrorID - Дополнительная информация об ошибке на стр. 61</i>
Data	ВХОД-ВЫХОД	ANY (любой)	Ссылка на данные для записи.
Axis	ВХОД-ВЫХОД	UDT 879	Ссылка на общие данные оси преобразователя частоты

## 7.10 FB 881 - VMC\_InitV1000\_RTU - Инициализация через Modbus RTU

### Описание

Этот блок используется для инициализации соответствующего преобразователя частоты пользовательскими данными. Он должен быть исполнен до начала подачи команд. Блок специально приспособлен для использования с преобразователями частоты при обмене с ними с использованием протокола Modbus RTU.



## Параметр

Параметр	Тип переменной	Тип данных	Описание
Execute	ВХОД	BOOL (Двоичное значение)	Задание запускается по переходу 0-1.
Hardware	ВХОД	BYTE	Спецификация используемого оборудования <ul style="list-style-type: none"> <li>1: Модуль SLIO CP040, логический адрес которого должен быть указан через <i>Laddr</i>.</li> <li>2: Процессорный модуль, выполненный на базе SPEED7</li> </ul>
Laddr	ВХОД	INT	Логический адрес модуля SLIO CP040 ( <i>Hardware</i> = 1). В противном случае этот параметр игнорируется.
UnitId	ВХОД	BYTE	Адрес Modbus для V1000.
UserUnitsVelocity	ВХОД	INT	Пользовательские единицы для скорости <ul style="list-style-type: none"> <li>0: Гц <ul style="list-style-type: none"> <li>– Задается в герцах</li> </ul> </li> <li>1: % <ul style="list-style-type: none"> <li>– Задается в процентах от максимальной скорости = <math>2 \cdot f_{\max} / p</math>, где <math>f_{\max}</math>: макс. выходная частота (параметр E1-04) p: Количество полюсов двигателя (индивидуальные параметры двигателя E2-04, E4-04 или E5-04)</li> </ul> </li> <li>2: RPM (об/мин) <ul style="list-style-type: none"> <li>– Данные в оборотах в минуту</li> </ul> </li> </ul>
UserUnitsAcceleration	ВХОД	INT	Пользовательские единицы для ускорения и замедления <ul style="list-style-type: none"> <li>0: 0,01 с (диапазон значений: 0,00 - 600,00 с)</li> <li>1: 0,1 с (диапазон значений: 0,0 - 6000,0 с)</li> </ul>
MaxVelocityApp	ВХОД	REAL	Максимальная скорость для приложения. Должна быть задана в пользовательских единицах. Используется в командах перемещения для калибровки.
Done	ВЫХОД	BOOL (Двоичное значение)	Status <ul style="list-style-type: none"> <li>TRUE: Задание успешно выполнено</li> </ul>
Busy	ВЫХОД	BOOL (Двоичное значение)	Status <ul style="list-style-type: none"> <li>TRUE: Задание выполняется</li> </ul>
Error	ВЫХОД	BOOL (Двоичное значение)	Status <ul style="list-style-type: none"> <li>TRUE: Возникла ошибка. Дополнительная информация об ошибке может быть взята из параметра <i>ErrorID</i>.</li> </ul>
ErrorID	ВЫХОД	WORD (16-разрядное значение)	Дополнительная информация об ошибке ↪ Раздел 8 'ErrorID - Дополнительная информация об ошибке на стр. 61
Axis	ВХОД-ВЫХОД	UDT 879	Ссылка на общие данные оси преобразователя частоты
V1000	ВХОД-ВЫХОД	UDT 881	Ссылка на пользовательские данные преобразователя частоты

## 7.11 FB 882 - VMC\_AxisControlV1000\_RTU - Управление осью через Modbus RTU

### Описание

С помощью блока FB 882 - VMC\_AxisControlV1000\_RTU можно управлять преобразователем частоты через Modbus RTU, а также контролировать его состояние.



Управление преобразователем частоты V1000, который подключается через Modbus RTU, осуществляется исключительно с помощью FB 882 VMC\_AxisControlV1000\_RTU. Блоки PLCopen не поддерживаются!

### Параметр

Параметр	Тип переменной	Тип данных	Описание
AxisEnabled	ВХОД	BOOL (Двоичное значение)	Активация оси <ul style="list-style-type: none"> <li>■ TRUE: Включить ось → <i>AxisEnabled</i> = 1, команды могут выполняться.</li> <li>■ FALSE: Выключить ось → <i>AxisEnabled</i> = 0, команды не могут выполняться.</li> </ul>
AxisReset	ВХОД	BOOL (Двоичное значение)	Команда: Сброс ошибки преобразователя частоты. → <i>CmdActive</i> = 1
StopExecute	ВХОД	BOOL (Двоичное значение)	Команда: Stop - Останов оси → <i>CmdActive</i> = 1
MvVelocityExecute	ВХОД	BOOL (Двоичное значение)	Команда: MoveVelocity (управление скоростью) → <i>CmdActive</i> = 2
Velocity	ВХОД	REAL	Параметр: Значение скорости для MoveVelocity в пользовательских единицах. См. пример ниже
AccelerationTime	ВХОД	REAL	Параметр: Время разгона в секундах (точность в зависимости от <i>UserUnitsAcceleration</i> в блоке инициализации Init). Всегда относится ко времени от состояния покоя до максимальной установленной скорости. См. пример ниже
DecelerationTime	ВХОД	REAL	Параметр: Время замедления в секундах (точность в зависимости от <i>UserUnitsAcceleration</i> в блоке инициализации Init). Всегда относится ко времени от максимальной установленной скорости до состояния покоя. См. пример ниже
JogPositive	ВХОД	BOOL (Двоичное значение)	Команда: JogPos <ul style="list-style-type: none"> <li>■ Переход 0-1: Начало движения оси в положительном направлении (толчковое перемещение вперёд)</li> <li>■ Переход 1-0: Останов оси</li> </ul>
JogNegative	ВХОД	BOOL (Двоичное значение)	Команда: JogNeg <ul style="list-style-type: none"> <li>■ Переход 0-1: Начало движения оси в отрицательном направлении (толчковое перемещение назад)</li> <li>■ Переход 1-0: Останов оси</li> </ul>
JogVelocity	ВХОД	REAL	Параметр: Значение скорости для толчкового режима в пользовательских единицах. См. пример ниже
JogAccelerationTime	ВХОД	REAL	Параметр: Время разгона в секундах (точность в зависимости от <i>UserUnitsAcceleration</i> в блоке инициализации Init). Всегда относится ко времени от состояния покоя до максимальной установленной скорости. См. пример ниже

Параметр	Тип переменной	Тип данных	Описание
JogDecelerationTime	ВХОД	REAL	Параметр: Время замедления в секундах (точность в зависимости от <i>UserUnitsAcceleration</i> в блоке FB 881). Всегда относится ко времени от максимальной установленной скорости до состояния покоя. См. пример ниже
AxisReady	ВЫХОД	BOOL (Двоичное значение)	Состояние: Готовность оси <ul style="list-style-type: none"> <li>■ TRUE: Ось готова к включению.</li> <li>■ FALSE: Ось не готова к включению.</li> </ul>
AxisEnabled	ВЫХОД	BOOL (Двоичное значение)	Состояние: Активация оси <ul style="list-style-type: none"> <li>■ TRUE: Ось включена</li> <li>■ FALSE: Ось выключена</li> </ul>
AxisError	ВЫХОД	BOOL (Двоичное значение)	Состояние: Ошибка оси <ul style="list-style-type: none"> <li>■ TRUE: Ось выдает ошибку и блокируется. Дополнительная информация об ошибке может быть взята из параметра <i>AxisErrorID</i>.</li> <li>■ FALSE: Ошибка оси отсутствует.</li> </ul>
AxisErrorID	ВЫХОД	WORD (16-разрядное значение)	Состояние: Дополнительная информация об ошибке для <i>AxisError</i> ↪ <i>Раздел 8 'ErrorID - Дополнительная информация об ошибке на стр. 61</i>
DriveError	ВЫХОД	BOOL (Двоичное значение)	Состояние: Ошибка преобразователя частоты <ul style="list-style-type: none"> <li>■ TRUE: Преобразователь частоты выдает ошибку и блокируется.</li> <li>■ FALSE: Ошибка преобразователя частоты отсутствует.</li> </ul>
ActualVelocity	ВЫХОД	REAL	Состояние: Текущая скорость в пользовательских единицах
InVelocity	ВЫХОД	BOOL (Двоичное значение)	Статус заданной скорости <ul style="list-style-type: none"> <li>■ TRUE: Заданная скорость <i>Velocity</i> достигнута.</li> <li>■ FALSE: Заданная скорость <i>Velocity</i> еще не достигнута.</li> </ul>
CmdDone	ВЫХОД	BOOL (Двоичное значение)	Состояние: Команда исполнена <ul style="list-style-type: none"> <li>■ TRUE: Команда исполнена успешно.</li> <li>■ FALSE: Команда еще не исполнена или всё ещё выполняется.</li> </ul>
CmdBusy	ВЫХОД	BOOL (Двоичное значение)	Состояние: Команда в процессе исполнения. <ul style="list-style-type: none"> <li>■ TRUE: Команда в процессе обработки.</li> <li>■ FALSE: В данный момент команда не выполняется.</li> </ul>
CmdAborted	ВЫХОД	BOOL (Двоичное значение)	Состояние: Команда прервана <ul style="list-style-type: none"> <li>■ TRUE: Команда была прервана.</li> <li>■ FALSE: Команда не была прервана.</li> </ul>
CmdError	ВЫХОД	BOOL (Двоичное значение)	Состояние: Ошибка команды <ul style="list-style-type: none"> <li>■ TRUE: Произошла ошибка при выполнении команды.</li> <li>■ FALSE: Команда выполнена без ошибок.</li> </ul>
CmdErrorID	ВЫХОД	WORD (16-разрядное значение)	Состояние: Дополнительная информация по <i>CmdError</i> ↪ <i>Раздел 8 'ErrorID - Дополнительная информация об ошибке' на стр. 61</i>

Параметр	Тип переменной	Тип данных	Описание
CmdActive	ВЫХОД	INT	Состояние: Активная команда <ul style="list-style-type: none"> <li>■ 0: NoCmd - нет активной команды</li> <li>■ 1: Stop</li> <li>■ 2: MvVelocity</li> <li>■ 3: MvRelative</li> <li>■ 4: JogPos</li> <li>■ 5: JogNeg</li> </ul>
DirectionPositive	ВЫХОД	BOOL (Двоичное значение)	Состояние: Направление вращения вперёд <ul style="list-style-type: none"> <li>■ TRUE: Текущее направление вращения вперёд</li> <li>■ FALSE: Текущее направление вращения не вперёд</li> </ul>
DirectionNegative	ВЫХОД	BOOL (Двоичное значение)	Состояние: Направление вращения назад <ul style="list-style-type: none"> <li>■ TRUE: Текущее направление вращения назад</li> <li>■ FALSE: Текущее направление вращения не назад</li> </ul>
Axis	ВХОД-ВЫХОД	UDT 879	Ссылка на общие данные оси преобразователя частоты
V1000	ВХОД-ВЫХОД	UDT 881	Ссылка на пользовательские данные преобразователя частоты
AxisComData	ВХОД-ВЫХОД	UDT 878	Ссылка на коммуникационные данные текущего ведомого устройства

### Пример для AccelerationTime

Значения для *Velocity*, *AccelerationTime* и *DecelerationTime* должны быть заданы в пользовательских единицах, заданных в FB 881 - VMC\_InitV1000\_RTU. *AccelerationTime* или *DecelerationTime* определяют время от состояния покоя и до максимальной установленной скорости или от максимальной установленной скорости и до состояния покоя.

Максимальная скорость определяется по формуле

$$v_{max} = \frac{2 \cdot f}{p}$$

$v_{max}$  - макс. скорость в 1/с

$f$  - макс. выходная частота (параметр E1-04)

$p$  - количество полюсов двигателя (индивидуальные параметры двигателя E2-04, E4-04 или E5-04)

### Последовательность действий

1. ➤ Выполните команду 'Project → Compile all' и загрузите проект в ЦПУ. Дополнительную информацию о процедуре загрузки можно найти в интерактивной справке по *SPEED7 Studio*.

⇒ Теперь можно запустить прикладную программу на исполнение.



#### ВНИМАНИЕ!

Всегда соблюдайте указания по технике безопасности для преобразователя частоты, особенно при вводе его в эксплуатацию!

2. ➤ Переведите ЦПУ в режим RUN и подайте питание на электропривод.

⇒ Блок FB 882 - VMC\_AxisControlV1000\_RTU обрабатывается циклически.

3. ➤ Как только *AxisReady* = TRUE, можно сделать доступным управление осью с помощью *AxisEnable*.

4. ➤ Теперь есть возможность управлять преобразователем частоты через соответствующие параметры и контролировать его состояние.

## 8 ErrorID - Дополнительная информация об ошибке

ErrorID	Описание	Примечание
0x0000	Ошибка отсутствует	
0x8y24	Ошибка в параметрах блока у, где у: <ul style="list-style-type: none"> <li>■ 1: Ошибка в PROTOKOLL</li> <li>■ 2: Ошибка в PARAMETER</li> <li>■ 3: Ошибка в BAUDRATE</li> <li>■ 4: Ошибка в CHARLENGTH</li> <li>■ 5: Ошибка в PARITY</li> <li>■ 6: Ошибка в STOPBITS</li> <li>■ 7: Ошибка в FLOWCONTROL (параметр отсутствует)</li> </ul>	VMC_ConfigMaster_RTU
0x8001	Недопустимое значение для параметра <i>Position</i> .	
0x8002	Недопустимое значение для параметра <i>Distance</i> .	
0x8003	Недопустимое значение для параметра <i>Velocity</i> .	
0x8004	Недопустимое значение для параметра <i>Acceleration</i> .	
0x8005	Недопустимое значение для параметра <i>Deceleration</i> .	
0x8007	Недопустимое значение для параметра <i>ContinuousUpdate</i> .	
0x8008	Недопустимое значение для параметра <i>BufferMode</i> .	
0x8009	Недопустимое значение для параметра <i>EnablePositive</i> .	
0x800A	Недопустимое значение для параметра <i>EnableNegative</i> .	
0x800B	Недопустимое значение для параметра <i>MasterOffset</i> .	
0x800C	Недопустимое значение для параметра <i>SlaveOffset</i> .	
0x800D	Недопустимое значение для параметра <i>MasterScaling</i> .	
0x800E	Недопустимое значение для параметра <i>SlaveScaling</i> .	
0x800F	Недопустимое значение для параметра <i>StartMode</i> .	
0x8010	Недопустимое значение для параметра <i>ActivationMode</i> .	
0x8011	Недопустимое значение для параметра <i>Source</i> .	
0x8012	Недопустимое значение для параметра <i>Direction</i> .	
0x8014	Недопустимый параметр физической оси.	Mc_ReadParameter
0x8015	Недопустимый индекс или субиндекс.	Mc_ReadParameter
0x8016	Недопустимая длина параметра.	Mc_ReadParameter
0x8017	Недопустимый LADDR.	Mc_ReadParameter
0x8018	Недопустимое значение для параметра <i>RatioDenominator</i> .	MC_GearIn
0x8019	Недопустимое значение для параметра <i>RatioNumerator</i> .	MC_GearIn
0x801A	Номер пааметра неизвестен.	Mc_ReadParameter, MC_WriteParameter
0x801B	Параметр не может быть записан, параметр защищен от записи.	MC_WriteParameter
0x801C	Связь параметров с неизвестным режимом.	MC_Home, MC_WriteParameter
0x801D	Связь параметров с общей ошибкой. Причина ошибки подробно не описана.	

ErrorID	Описание	Примечание
0x801E	Значение параметра SDO вне диапазона.	MC_Home, MC_WriteParameter
0x801F	Тип в ANY не BYTE.	Параметр для чтения/записи
0x8020	Разная конфигурация пользовательских блоков для кулачка (CAM) и главной оси.	
0x8021	Разная конфигурация пользовательских блоков для кулачка (CAM) и ведомой оси.	
0x8022	По логическому адресу, указанному через LADDR, нет устройства PROFIBUS / PROFINET, из которого можно считывать согласованные данные.	Параметр для чтения/записи
0x8023	При обращении к устройству ввода-вывода была обнаружена ошибка доступа.	Параметр для чтения/записи
0x8024	Ошибка ведомого во внешнем ведомом устройстве PROFIBUS DP.	Параметр для чтения/записи
0x8025	Системная ошибка во внешнем ведомом устройстве PROFIBUS DP.	Параметр для чтения/записи
0x8026	Системная ошибка во внешнем ведомом устройстве PROFIBUS DP.	Параметр для чтения/записи
0x8027	Данные еще не были прочитаны модулем.	Параметр для чтения/записи
0x8028	Системная ошибка во внешнем ведомом устройстве PROFIBUS DP.	Параметр для чтения/записи
0x8029	Попытка записи в объект, который доступен только по чтению.	Параметр для чтения/записи
0x802A	Попытка чтения из объекта, который доступен только по записи.	Параметр для чтения/записи
0x802B	Неподдерживаемый доступ к объекту.	Параметр для чтения/записи
0x802C	Неверный тип данных.	Параметр для чтения/записи
0x802D	Ошибка в профиле устройства.	Параметр для чтения/записи
0x802E	Тип команды ошибки.	Параметр для чтения/записи
0x802F	Нет доступных системных ресурсов.	Параметр для чтения/записи
0x8030	Недопустимое значение для параметра <i>Hardware</i> (1 = SLIO CP, 2 = CPU VIPA).	Modbus; Init
0x8031	Недопустимое значение для параметра <i>UnitId</i> .	Modbus; Init
0x8032	Недопустимое значение для параметра <i>UserUnitsVelocity</i> (0 = Гц, 1 = %, 2 = об/мин (RPM)).	Modbus; Init
0x8033	Недопустимое значение для параметра <i>UserUnitsAcceleration</i> (0 = 0,00 с, 1 = 0,0 с).	Modbus; Init
0x8034	Недопустимое значение для параметра <i>MaxVelocityApp</i> (должно быть > 0).	Modbus; Init
0x8035	Ошибка доступа к <i>MonitorData</i> при чтении.	Modbus; Init
0x8036	Ошибка доступа к <i>NumberOfPoles</i> при чтении.	Modbus; Init
0x8037	Ошибка доступа к <i>UserUnitsVelocity</i> при записи.	Modbus; Init
0x8038	Ошибка доступа к <i>MinOutputFrequency</i> при чтении.	Modbus; Init
0x8039	Ошибка доступа к <i>MaxOutputFrequency</i> при чтении.	Modbus; Init
0x803A	Ошибка доступа к <i>StoppingMethodSelection</i> при записи.	Modbus; Init
0x803B	Ошибка доступа к <i>UserUnitsAcceleration</i> при записи.	Modbus; Init
0x8041	Недопустимое значение для параметра <i>AccelerationTime</i> .	Modbus V1000
0x8042	Недопустимое значение для параметра <i>DecelerationTime</i> .	Modbus V1000
0x8043	Недопустимое значение для параметра <i>JogAccelerationTime</i> .	Modbus V1000
0x8044	Недопустимое значение для параметра <i>JogDecelerationTime</i> .	Modbus V1000

ErrorID	Описание	Примечание
0x8045	Недопустимое значение для параметра <i>JogVelocity</i> ( $\leq MaxVelocityApp$ ).	Modbus V1000
0x80C8	Коммуникационная ошибка Modbus: отсутствие ответа от сервера в течение определенного периода времени (таймаут может быть параметризован через интерфейс).	Modbus V1000
0x809y	Ошибка в значении параметра блока y, где y: <ul style="list-style-type: none"> <li>■ 1: Ошибка в PROTOKOLL</li> <li>■ 3: Ошибка в BAUDRATE</li> <li>■ 4: Ошибка в CHARLENGTH</li> <li>■ 5: Ошибка в PARITY</li> <li>■ 6: Ошибка в STOPBITS</li> </ul>	VMC_ConfigMaster_RTU
0x8092	Ошибка доступа к параметру DB (слишком короткий DB).	VMC_ConfigMaster_RTU
0x809A	Интерфейс недоступен или работает в режиме PROFIBUS.	VMC_ConfigMaster_RTU
0x8101	Циклический обмен данными с осью невозможен.	
0x8102	PLCopen-состояние не определено.	
0x8103	Команда не поддерживается осью.	
0x8104	Ось не готова к включению, возможные причины: <ul style="list-style-type: none"> <li>■ Связь с осью не готова.</li> <li>■ Привод не находится в состоянии 'switched on' → сброс ошибки привода возможен с помощью MC_Reset.</li> <li>■ Связь была прервана, например, из-за выключения ЦПУ. Сброс ошибки с помощью MC_Reset.</li> </ul>	<i>PreOperational</i> также должен быть установлен в <i>Operational</i> .
0x8105	Команда не поддерживается виртуальной осью.	
0x8106	PLCopen-состояние не определено.	
0x8107	Команда не разрешена, когда привод отключен.	VMC_AxisControl_PT, Mod- busV1000
0x8188	Коммуникационная ошибка Modbus: Внутренняя ошибка MB_FUNCTION недействительна.	Modbus V1000
0x8189	Коммуникационная ошибка Modbus: Внутренняя ошибка MB_DATA_ADDR недействительна.	Modbus V1000
0x818A	Коммуникационная ошибка Modbus: Внутренняя ошибка MB_DATA_LEN недействительна.	Modbus V1000
0x818B	Коммуникационная ошибка Modbus: Внутренняя ошибка MB_DATA_PTR недействительна.	Modbus V1000
0x8201	Команда не может быть выполнена в настоящее время из-за отсутствия внутренних ресурсов (без свободного слота в CommandBuffer).	
0x8202	Ошибка записи смещения для Homing (отсутствие свободного слота в CommandBuffer).	DriveManager → Homing (активная команда)
0x8210	Коммуникационная ошибка Modbus: Аппаратное обеспечение несовместимо с библиотекой блоков Modbus RTU/TCP.	Modbus V1000
0x828y	Ошибка в параметре y DB параметров, где y: <ul style="list-style-type: none"> <li>■ 1: Ошибка в параметре №1</li> <li>■ 2: Ошибка в параметре №2</li> <li>■ ...</li> </ul>	VMC_ConfigMaster_RTU
0x8301	Циклический обмен данными с ведущей осью невозможен.	

ErrorID	Описание	Примечание
0x8302	PLCореп-состояние не определено.	
0x8303	Команда не поддерживается ведущей осью.	
0x8304	Ведущая ось не находится в состоянии Pre-Operational.	
0x8305	Изменен номер блока данных ведущей оси.	
0x8306	Ошибка связи с ведущей осью. Быстрый останов ведомой оси.	
0x8311	Циклический обмен данными с ведомой осью невозможен.	
0x8312	Недопустимая команда для текущего PLCореп-состояния ведомой оси.	
0x8313	Команда не поддерживается ведомой осью.	
0x8314	Ведущая ось не находится в состоянии <i>Pre-Operational</i> .	
0x8315	Изменен номер блока данных ведущей оси.	
0x8321	Сопряжение с помощью <i>StartMode = relative</i> и <i>ActivationMode = nextcycle</i> не допускается.	
0x8322	Сопряжение с помощью <i>StartMode = absolute</i> и <i>ActivationMode = nextcycle</i> не допускается.	
0x8323	Переключение с разными <i>StartMode</i> (должен использоваться <i>StartMode</i> муфты)	
0x8331	MC_CamIn неактивен.	
0x8332	MC_GearIn неактивен.	
0x8340	Недопустимое значение в TriggerInput.Probe.	MC_TouchProbe и MC_AbortTrigger
0x8341	Недопустимое значение в TriggerInput.Source.	MC_TouchProbe и MC_AbortTrigger
0x8342	Недопустимое значение в TriggerInput.TriggerMode.	MC_TouchProbe и MC_AbortTrigger
0x8350	Недопустимое значение в VelocitySearchSwitch.	Homing, инициализация
0x8351	Недопустимое значение в VelocitySearchZero.	Homing, инициализация
0x8352	Недопустимая комбинация входов.	Homing, инициализация
0x8360	ЦПУ не поддерживает режим Pulse Train.	VMC_AxisControl_PT
0x8361	Неверное значение в S_ChannelNumberPWM.	VMC_AxisControl_PT
0x8362	Общая ошибка выхода Pulse Train.	VMC_AxisControl_PT
0x8363	Команда движения с набором StopExecute.	VMC_AxisControl_PT, Mod- busV1000
0x8381	Коммуникационная ошибка Modbus: сервер возвращает код исключения 01h.	Modbus V1000
0x8382	Коммуникационная ошибка Modbus: сервер возвращает код исключения 03h или или неправильный стартовый адрес.	Modbus V1000
0x8383	Коммуникационная ошибка Modbus: сервер возвращает код исключения 02h.	Modbus V1000
0x8384	Коммуникационная ошибка Modbus: сервер возвращает код исключения 04h.	Modbus V1000



ErrorID	Описание	Примечание
0x8386	Коммуникационная ошибка Modbus: сервер возвращает неверный код функции.	Modbus V1000
0x8388	Коммуникационная ошибка Modbus: Сервер возвращает ошибочное значение или ошибочный номер.	Modbus V1000
0x8400	MC_Power: непредусмотренное состояние привода Состояние привода <> Работа разрешена	MC_Power
0x8401	MC_Power: непредусмотренное состояние привода Состояние привода = Активен быстрый останов	MC_Power
0x8402	MC_Power: непредусмотренное состояние привода Состояние привода = Активна реакция на ошибку	MC_Power
0x8403	MC_Power: непредусмотренное состояние привода Состояние привода = Ошибка	MC_Power
0x8410	Тайм-аут при попытке сброса привода.	Базовый FB --> MC_Reset
0x8500	Неверное значение в <i>EncoderType</i> (1 или 2).	Блок инициализации
0x8501	Неверное значение в <i>EncoderResolutionBits</i> (>0 и $\leq 32$ ).	Блок инициализации
0x8502	Неверное значение в <i>LogicalAddress</i> ( $\geq 0$ ).	Блок инициализации
0x8503	Неверное значение в <i>StartInputAddress</i> ( $\geq 0$ ).	Блок инициализации
0x8504	Неверное значение в <i>StartOutputAddress</i> ( $\geq 0$ ).	Блок инициализации
0x8505	Неверное значение в <i>FactorPosition</i> (>0.0).	Блок инициализации
0x8506	Неверное значение в <i>FactorVelocity</i> (>0.0).	Блок инициализации
0x8507	Неверное значение в <i>FactorAcceleration</i> (>0.0).	Блок инициализации
0x8508	Неверное значение в <i>MaxVelocityApp</i> (>0.0).	Блок инициализации
0x8509	Неверное значение в <i>MaxAccelerationApp</i> (>0.0).	Блок инициализации
0x850A	Неверное значение в <i>MaxDecelerationApp</i> (>0.0).	Блок инициализации
0x850B	Неверное значение в <i>MaxVelocityDrive</i> (>0.0).	Блок инициализации
0x850C	Неверное значение в <i>MaxAccelerationDrive</i> (>0.0).	Блок инициализации
0x850D	Неверное значение в <i>MaxDecelerationDrive</i> (>0.0).	Блок инициализации
0x850E	Неверное значение в <i>MinPosition</i> ( $\geq$ MinUserPos).	Блок инициализации
0x850F	Неверное значение в <i>MaxPosition</i> ( $\geq$ MaxUserPos).	Блок инициализации
0x8510	Неверное значение в <i>M2_EncoderType</i> .	VMC_InitSigma7W_EC
0x8511	Неверное значение в <i>M2_EncoderResolutionBits</i>	VMC_InitSigma7W_EC
0x8513	Неверное значение в <i>M2_PdoInputs</i> .	VMC_InitSigma7W_EC
0x8514	Неверное значение в <i>M2_PdoOutputs</i> .	VMC_InitSigma7W_EC
0x8515	Неверное значение в <i>M2_FactorPosition</i> .	VMC_InitSigma7W_EC
0x8516	Неверное значение в <i>M2_FactorVelocity</i> .	VMC_InitSigma7W_EC
0x8517	Неверное значение в <i>M2_FactorAcceleration</i> .	VMC_InitSigma7W_EC
0x8518	Неверное значение в <i>M2_MaxVelocityApp</i> .	VMC_InitSigma7W_EC
0x8519	Неверное значение в <i>M2_MaxAccelerationApp</i> .	VMC_InitSigma7W_EC
0x851A	Неверное значение в <i>M2_MaxDecelerationApp</i> .	VMC_InitSigma7W_EC
0x8603	Ошибка Homing в приводе, скорость <> 0.	MC_Home
0x8604	Ошибка Homing в приводе, скорость = 0.	MC_Home

ErrorID	Описание	Примечание
0x8700	Ошибка: Недопустимый размер.	
0x8710	Ошибка SDO: Бит переключения не был изменен.	
0x8711	Ошибка SDO: Тайм-аут протокола SDO.	
0x8712	Ошибка SDO: Команда "клиент/сервер" недействительна или неизвестна.	
0x8713	Ошибка SDO: Недопустимый размер блока (только в блочном режиме).	
0x8714	Ошибка SDO: Недопустимый порядковый номер (только в блочном режиме).	
0x8715	Ошибка SDO: Ошибка CRC (только в режиме блока).	
0x8716	Ошибка SDO: Недостаточно памяти.	
0x8717	Ошибка SDO: Неподдерживаемый доступ к объекту.	
0x8718	Ошибка SDO: Попытка чтения из объекта, который доступен только по записи.	
0x8719	Ошибка SDO: Попытка записи в объект, который доступен только по чтению.	
0x871A	Ошибка SDO: Объект не существует в словаре объектов.	
0x871B	Ошибка SDO: Объект не может быть сопоставлен с PDO.	
0x871C	Ошибка SDO: Количество и длина объектов, подлежащих отображению, превышают длину PDO.	
0x871D	Ошибка SDO: Общая несовместимость параметров.	
0x871E	Ошибка SDO: Общая внутренняя несовместимость в устройстве.	
0x871F	Ошибка SDO: Ошибка доступа из-за сбоя оборудования.	
0x8720	Ошибка SDO: Тип данных не соответствует, длина служебного параметра не соответствует.	
0x8721	Ошибка SDO: Тип данных не соответствует, служебный параметр слишком длинный.	
0x8722	Ошибка SDO: Тип данных не соответствует, служебный параметр слишком длинный.	
0x8723	Ошибка SDO: Субиндекс отсутствует.	
0x8724	Ошибка SDO: Доступ по записи - Значение параметра вне диапазона.	
0x8725	Ошибка SDO: Доступ по записи - Значение параметра выше верхней границы диапазона.	
0x8726	Ошибка SDO: Доступ по записи - Значение параметра ниже нижней границы диапазона.	
0x8727	Ошибка SDO: Максимальное значение < Минимальное значение.	
0x8728	Ошибка SDO: Общая ошибка.	
0x8729	Ошибка SDO: Данные не могут быть переданы в приложение или сохранены там.	
0x872A	Ошибка SDO: Данные не могут быть переданы в приложение или сохранены там, потому что локальное управление включено.	
0x872B	Ошибка SDO: Из-за текущего состояния устройства никакие данные не могут быть переданы в приложение или сохранены там.	
0x872C	Ошибка SDO: Динамическая генерация каталога объектов не может быть выполнена или каталог объектов не существует.	

ErrorID	Описание	Примечание
0x872D	Ошибка SDO: Неизвестный код.	
0x8750	Неверное значение в LADDR.	
0x8751	Тип указателя ANY отличается от BYTE.	
0x8752	По адресу, указанному через LADDR, нет модуля PROFIBUS DP или устройства PROFINET IO, из которого можно считывать согласованные данные.	
0x8753	Ошибка доступа при обращении к устройству PROFINET IO.	
0x8754	Ошибка ведомого на внешнем ведомом PROFIBUS DP.	
0x8755	Длина данных SFB не соответствует длине пользовательских данных.	
0x8756	Ошибка на внешнем ведомом PROFIBUS DP.	
0x8757	Системная ошибка на внешнем ведомом PROFIBUS DP.	
0x8758	Данные еще не были прочитаны устройством.	
0x8759	Системная ошибка на внешнем ведомом PROFIBUS DP.	
0x875A	Системные ресурсы отсутствуют.	
0x8799	Ошибка SDO: Произошла другая ошибка, более подробную информацию см. в <i>Info1</i> и <i>Info2</i> .	
0x8888	Внутренняя: ошибка BufferIndex	VMC_AxisControl_PT
0xC000	Внутренняя ошибка: Статус Init не определен.	Modbus; Init
0xC001	Внутренняя ошибка: Недопустимое значение для параметра <i>Cmd.ActiveType</i> .	Modbus V1000
0xC002	Внутренняя ошибка: Недопустимое значение для параметра <i>Cmd.State</i> .	Modbus V1000