



# Управление электроприводом Микромастер 440 по сети PROFIBUS

Учебно-методическое пособие

## Автор

Романов В. П. , преподаватель НОУ «РЦПП «Евраз-Сибирь», преподаватель высшей квалификационной категории.

## Рецензент

Зав. Кафедрой «Автоматизированного электропривода и промышленной электроники» ФГОУ ВПО «СибГИУ» доктор технических наук, профессор  
Островлянчик Ю. В.

В пособии изложены основы организации связи с использованием промышленной шины PROFIBUS, рассмотрены принципы управления частотным электроприводом Micromaster 440 с использованием шины PROFIBUS, представлены примеры программ для систем управления частотным электроприводом Micromaster 440 на основе PLC фирмы Siemens с использованием шины PROFIBUS.

Предназначено для слушателей курсов целевого назначения электромонтеров по ремонту и обслуживанию электрооборудования.

Зам. директора НОУ «РЦПП «Евраз-Сибирь»

Н. В. Пыжкова

# Оглавление

ВВЕДЕНИЕ.....	4
1. Основы PROFIBUS.....	5
1.1. Модель ISO/OSI.....	5
1.2. Функции уровней модели ISO/OSI.....	6
1.3. Архитектура протоколов и профили PROFIBUS.....	7
1.3.1. PROFIBUS-DP.....	7
1.3.2. PROFIBUS-FMS.....	7
1.3.3. PROFIBUS-PA.....	8
1.4. Уровни PROFIBUS-DP.....	8
1.4.1. Физический уровень (Layer 1) для DP/FMS (RS485).....	8
1.4.1.1. Способ передачи.....	8
1.4.2. Канальный уровень (Уровень 2).....	9
1.4.2.1. Шинная топология.....	9
1.4.2.2. Подключение шины.....	10
1.4.2.3. Окончание шины.....	10
1.5. Пользовательский интерфейс DP и DP-профили.....	11
1.6. Управление доступом к шине в PROFIBUS.....	11
1.6.1. Метод обмена маркером.....	12
1.6.2. Метод Master-Slave.....	13
1.7. Цикл PROFIBUS-DP.....	13
1.7.1. Структура циклов PROFIBUS.....	13
1.7.2. Структура постоянного по времени PROFIBUS-DP цикла.....	14
1.7.3. Функциональное расширение DP-V1.....	15
2. Взаимодействие с MICROMASTER4 через PROFIBUS-DP.....	15
2.1. Подключение PROFIBUS-DP к преобразователям частоты и установка адреса.....	15
2.2. Структура рабочих данных согласно профилю PROFIDrive 2.0 и 3.0.....	17
2.2.1. Слово управления/состояния и сигнал задания на скорость.....	18
2.2.2. Механизм PKW для обработки параметров.....	21
2.2.3. Параметрирование преобразователя для работы в сети PROFIBUS-DP.....	25
3. PROFIBUS-DP в системе SIMATIC S7.....	26
3.1. DP-интерфейсы в системах SIMATIC S7.....	26
3.1.1. Свойства запуска интерфейса DP-Master в SIMATIC S7.....	27
3.1.2. Выход из строя станции DP-Slave.....	27
3.1.3. Сигнал (Alarm) при удалении/вставке модуля.....	27
3.1.4. Диагностические сигналы от станций DP-Slave.....	27
3.1.5. Сигналы от процесса у станций DP-Slave.....	27
3.1.6. Сигнал состояния (Statusalarm) от DP-Slave'a.....	28
4. Краткая информация о программном обеспечении STEP 7.....	28
4.1. Общие сведения о программном обеспечении STEP 7.....	28
4.2. Основы STEP 7.....	28
4.2.1. Структура программы.....	28
5. Пользовательское программирование DP-интерфейса.....	36
5.1. Пример программы управления.....	36
6. Функции для обмена DP-пользовательскими данными и функции для сигналов от процесса.....	40
6.1. Обмен консистентными DP-данными с помощью SFC14 DPRD_DAT и SFC15 DPWR_DAT.....	40
6.2. Пример программы управления.....	41
6.3. Пример программы управления с использованием свободной конфигурация циклических данных.....	46
6.3.1. Определение адресации устройств на шине Profibus.....	46
6.3.2. Конфигурирования параметров связи для преобразователя.....	46
6.3.3. Определения необходимых параметров требуемых для контроля ПЧ при проектировании исходящей информации от преобразователя.....	46
6.3.4. Проектирование использования входящей информации в преобразователе.....	48
6.3.5. Конфигурирование и программирование ПЛК с использованием Step7.....	49
6.3.6. Разработка функции обмена данными с ПЧ в Step7.....	49
Литература.....	60

## **ВВЕДЕНИЕ**

В настоящее время автоматизированные системы управления технологическими процессами получили широкое применение на промышленных предприятиях. Это в первую очередь связано с возрастающими требованиями к качеству выпускаемой продукции и гибкости производственного процесса. Такие системы включают в себя несколько уровней автоматизации, начиная с систем визуализации или систем человеко-машинного интерфейса на верхних уровнях и заканчивая исполнительными устройствами, электроприводами и датчиками различных переменных на нижних уровнях. Взаимодействие между системами визуализации и электроприводами в сложных автоматизированных системах осуществляется посредством логических контроллеров.

Для обмена информацией между средствами автоматизации, которые в общем случае могут быть удалены друг от друга на значительные расстояния, применяются промышленные информационные сетевые технологии. Одной из наиболее известных и часто используемых информационных технологий является промышленная сеть PROFIBUS.

В исполнительных приводных устройствах доминирующее положение занимают частотно-регулируемые электроприводы переменного тока. Это обусловлено широкой распространенностью, простотой конструкции и высокой надежностью асинхронного двигателя.

В учебно-методическое пособие рассмотрены принципы управления частотным электроприводом Micromaster 440 с использованием шины PROFIBUS, представлены примеры программ для систем управления частотным электроприводом Micromaster 440 на основе PLC фирмы Siemens с использованием шины PROFIBUS.

Предназначено для слушателей курсов целевого назначения электромонтеров по ремонту и обслуживанию электрооборудования.

# 1. Основы PROFIBUS

PROFIBUS (**PRO**cess **FI**eld **BUS**) представляет собой международный, открытый стандарт полевых шин с широким диапазоном применения в автоматизации технологических и производственных процессов. Независимость от производителя и открытость стандарта гарантируются международными нормами IEC 61158 или IEC 61158.

PROFIBUS-DP может выступать в качестве замены как обычной, параллельной передачи сигналов 24 В в системах автоматизации производства, так и для передачи сигналов в аналоговом виде 4...20 мА в системах автоматизации технологических процессов.

## 1.1. Модель ISO/OSI

Архитектура протоколов PROFIBUS ориентирована на уже установленные национальные и международные нормы. Так, архитектура протоколов базируется на модели OSI (Open System Interconnection).

На рис.1 изображена модель ISO/OSI для коммуникационных стандартов, состоящая из 7 уровней, подразделяющихся на два класса:

- ориентированных на пользователя с уровня 5 по уровень 7;
- ориентированных на сеть (уровни 1-4).

Уровни с 1 по 4 описывают пересылку передаваемых данных из одного пункта в другой, в то время как уровни с 5 по 7 предоставляют в распоряжение пользователя доступ к сети в соответствующей форме.

Номер уровня	Обозначение	Краткое описание	
7	Прикладной (Application)	Прикладной уровень отвечает за передачу информации от интерфейса приложения к любому сетевому узлу. Протоколы на этом уровне сильно различаются по размеру и сложности	↑ <b>Ориентированы на пользователя</b> ↓
6	Представления (Presentation)	Отвечает за представление данных в удобном для пользователя виде. На этом уровне выполняется сжатие/восстановление и шифрование/дешифрование данных	
5	Сеансовый (Session)	Предназначен для поддержки двух верхних уровней. На данном уровне устанавливается метод связи между удаленными системами – удаленный вызов процедур (RPC). Включает в себя две функции: управление диалогом и разделение данных	
4	Транспортный (Transport)	Протоколы транспортного уровня, такие как, например TCP, обеспечивают передачу данных по логическим адресам, которые определяются на предыдущем сетевом уровне	↑ <b>Ориентированы на сеть</b> ↓
3	Сетевой (Network)	На этом уровне протоколы отвечают за определение наилучшего пути маршрутизации данных. Здесь определяются логические сетевые адреса, например, такие как IP. Протоколы не отвечают за доставку по логическим адресам, а только обеспечивают наилучший путь данных	
2	Канальный (Data Link)	Протоколы на этом уровне должны обеспечить безошибочную передачу данных. На этом уровне битовый поток собирается в кадры (пакеты) в которые включается адресная информация. Здесь обеспечивается контроль ошибок и механизм повторной передачи пакетов	
1	Физический (Physical)	Описывает физическую среду передачи данных: медные провода, оптоволокно и т.д. На физическом уровне данные представляются в виде последовательности битов, без информации о формировании кадра. На физическом уровне не предусмотрено никакой адресации	

Рисунок 1 - Уровни модели ISO/OSI для протокола

## 1.2. Функции уровней модели ISO/OSI

**Физический уровень.** Этот уровень имеет дело с передачей битов по физическим каналам, таким, например, как коаксиальный кабель, витая пара или оптоволоконный кабель. К этому уровню имеют отношение характеристики физических сред передачи данных, такие как полоса пропускания, помехозащищенность, волновое сопротивление и другие. На этом же уровне определяются характеристики электрических сигналов, такие как требования к фронтам импульсов, уровням напряжения или тока передаваемого сигнала, тип кодирования, скорость передачи сигналов. Кроме этого, здесь стандартизуются типы разъемов и назначение каждого контакта.

**Канальный уровень.** На физическом уровне просто пересылаются биты. При этом не учитывается, что в некоторых сетях, в которых линии связи используются (разделяются) попеременно несколькими парами взаимодействующих компьютеров, физическая среда передачи может быть занята. Поэтому одной из задач канального уровня является проверка доступности среды передачи. Другой задачей канального уровня является реализация механизмов обнаружения и коррекции ошибок. Для этого на канальном уровне биты группируются в наборы, называемые кадрами (frames). Канальный уровень обеспечивает корректность передачи каждого кадра, помещая специальную последовательность бит в начало и конец каждого кадра, чтобы отметить его, а также вычисляет контрольную сумму, суммируя все байты кадра определенным способом и добавляя контрольную сумму к кадру. Когда кадр приходит, получатель снова вычисляет контрольную сумму полученных данных и сравнивает результат с контрольной суммой из кадра. Если они совпадают, кадр считается правильным и принимается. Если же контрольные суммы не совпадают, то фиксируется ошибка.

**Сетевой уровень.** Этот уровень служит для образования единой транспортной системы, объединяющей несколько сетей с различными принципами передачи информации между конечными узлами. Рассмотрим функции сетевого уровня на примере локальных сетей. Протокол канального уровня локальных сетей обеспечивает доставку данных между любыми узлами только в сети с соответствующей типовой топологией.

Таким образом, внутри сети доставка данных регулируется канальным уровнем, а вот доставкой данных между сетями занимается сетевой уровень.

Сообщения сетевого уровня принято называть пакетами (packets). При организации доставки пакетов на сетевом уровне используется понятие "номер сети". В этом случае адрес получателя состоит из номера сети и номера компьютера в этой сети.

**Транспортный уровень.** На пути от отправителя к получателю пакеты могут быть искажены или утеряны. Хотя некоторые приложения имеют собственные средства обработки ошибок, существуют и такие, которые предпочитают сразу иметь дело с надежным соединением. Работа транспортного уровня заключается в том, чтобы обеспечить приложениям или верхним уровням стека - прикладному и сеансовому - передачу данных с той степенью надежности, которая им требуется. Модель OSI определяет пять классов сервиса, предоставляемых транспортным уровнем. Эти виды сервиса отличаются качеством предоставляемых услуг: срочностью, возможностью восстановления прерванной связи, наличием средств мультиплексирования нескольких соединений между различными прикладными протоколами через общий транспортный протокол, а главное - способностью к обнаружению и исправлению ошибок передачи, таких как искажение, потеря и дублирование пакетов.

**Сеансовый уровень.** Сеансовый уровень обеспечивает управление диалогом для того, чтобы фиксировать, какая из сторон является активной в настоящий момент, а также предоставляет средства синхронизации. Последние позволяют вставлять контрольные точки в длинные передачи, чтобы в случае отказа можно было вернуться назад к последней контрольной точке, вместо того, чтобы начинать все с начала. На практике немногие приложения используют сеансовый уровень, и он редко реализуется.

**Уровень представления.** Этот уровень обеспечивает гарантию того, что информация, передаваемая прикладным уровнем, будет понятна прикладному уровню в другой системе. При необходимости уровень представления выполняет преобразование форматов данных в некоторый

общий формат представления, а на приеме, соответственно, выполняет обратное преобразование.

**Прикладной уровень.** Прикладной уровень - это в действительности просто набор разнообразных протоколов, с помощью которых пользователи сети получают доступ к разделяемым ресурсам, таким как файлы, принтеры или гипертекстовые Web-страницы, а также организуют свою совместную работу, например, с помощью протокола электронной почты.

### 1.3. Архитектура протоколов и профили PROFIBUS

Из рисунка 2, представляющего архитектуру протоколов PROFIBUS, можно видеть, что в ней реализованы уровни 1,2 и 7. Для уровней 1 и 2 принят стандарт США EIA (Electronic Industries Association) RS485, международные нормы IEC 870-5-1 (Telecontrol Equipment and System) и EN 60870-5-1. Метод доступа к сети, службы передачи и управления данными ориентируются на DIN 19241, части 1-3 и нормы IEC 955 Process Data Highway/Typ C. Функции управления (FMA7) ориентированы на ISO DIS 7498-4 (Management Framework).

С точки зрения пользователя PROFIBUS подразделяется на 3 профиля протокола: DP, FMS и PA.

Layer 7 Прикладной	<u>PNO-профиль для DP- устройств</u>	<u>PNO-профиль для FMS- устройств</u>	<u>PNO-профиль для DP- устройств</u>
	Основные функции Расширенные функции		Основные функции Расширенные функции
	DP User Interface Direct Data Link Mapper (DDLМ)	Application Layer Interface (ALI)	DP User Interface Direct Data Link Mapper (DDLМ)
Layer 3-6	НЕ ИСПОЛЬЗУЮТСЯ		
Layer 2 Канальный	Data Link Layer Fieldbus Data Link (FDL)	Data Link Layer Fieldbus Data Link (FDL)	IEC-Interface
Layer 1 Физический	Physical-Layer (RS485/LWL)	Physical-Layer (RS485/LWL)	IEC 1158-2

Рисунок 2 - Архитектура протокола PROFIBUS

#### 1.3.1. PROFIBUS-DP

PROFIBUS-DP применяет уровни 1 и 2, а также пользовательский интерфейс. Уровни с 3 по 7 не используются. Благодаря такой архитектуре достигается быстрая передача данных. Direct Data Link Mapper (DDLМ) организует доступ к уровню 2. В основу пользовательского интерфейса положены необходимые пользовательские функции, а также системные и аппаратно-зависимые функции различных типов PROFIBUS-DP-приборов.

Этот профиль протокола PROFIBUS оптимизирован для быстрого обмена данными специально для коммуникаций между системами автоматизации и децентрализованной периферией на полевом уровне.

#### 1.3.2. PROFIBUS-FMS

В PROFIBUS-FMS применяются уровни 1,2 и 7. Пользовательский уровень состоит из FMS (Fieldbus Message Specification) и LLI (Lower Layer Interface). FMS содержит пользовательский протокол и предоставляет в распоряжение коммуникационные службы.

LLI реализует различные коммуникационные связи и создает для FMS аппаратно-независимый доступ к уровню 2.

FMS применяется для обмена данными на уровне ячеек (PLC и PC). Мощные FMS-сервисы открывают широкие области использования и большую гибкость при передаче больших объемов



данных. PROFIBUS-DP и PROFIBUS-FMS применяют одинаковую технику передачи и единый протокол доступа к шине и поэтому могут работать через общий кабель.

### 1.3.3. PROFIBUS-PA

PROFIBUS-PA применяет расширенный PROFIBUS-DP-протокол передачи данных. Техника передачи согласно IEC 1158-2 обеспечивает надежность и питание полевых приборов через шину. Приборы PROFIBUS-PA могут благодаря применению специальных устройств (PROFIBUS-PA-Links) в простейшем случае интегрироваться в PROFIBUS-DP-сеть.

PROFIBUS-PA – специальная концепция, позволяющая подключать к общей шине датчики и приводы, находящиеся во взрывоопасной зоне.

## 1.4. Уровни PROFIBUS-DP

### 1.4.1. Физический уровень (Layer 1) для DP/FMS (RS485)

На физическом уровне в качестве среды передачи данных могут использоваться следующие технологии:

- экранированная витая пара (RS 485);
- искробезопасная экранированная витая пара для PROFIBUS-PA;
- волоконно-оптический кабель;
- беспроводные сети (ИК-технология).

На практике наибольшее распространение получила экранированная витая пара (рис. 3), с помощью которой можно передавать данные на расстояния до 12 км со скоростью передачи от 9,6 Кбит/с (12 км) до 12 Мбит/с (100 м). Скорость передачи по экранированной витой паре зависит от расстояния.

Кабели PROFIBUS имеют достаточно много различных исполнений для конкретных применений, например гирляндного исполнения, для подвижных механизмов, для прокладки под землей и др.

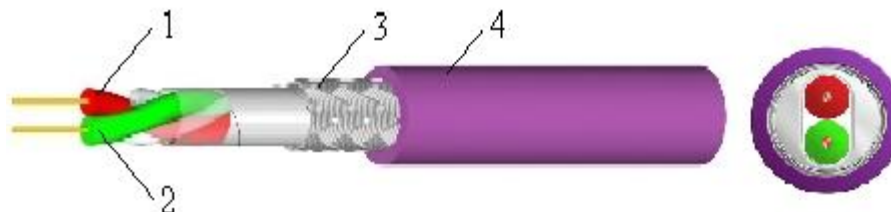


Рисунок 3 - Стандартный кабель PROFIBUS-DP:

1 – проводник А; 2 – проводник В; 3 – экран; 4 – изоляция

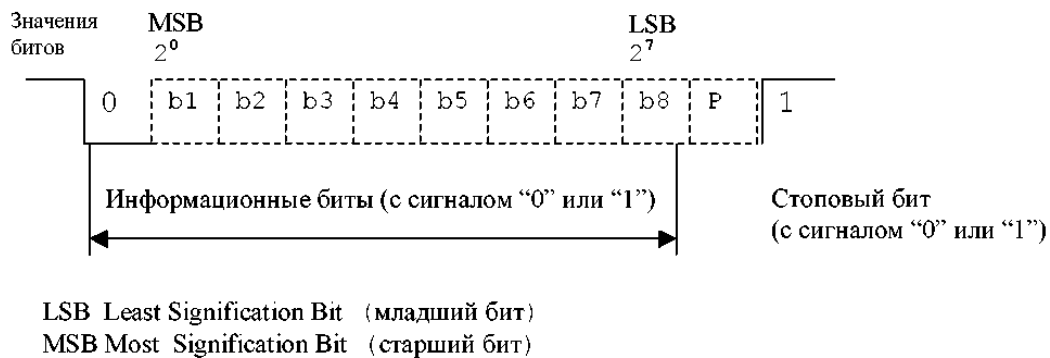
В основной версии для экранированной витой пары уровню 1 PROFIBUS соответствует симметричная передача данных по стандарту EIA RS485 (также обозначается H2). Проводники шинных сегментов замкнуты с обеих сторон, скручены и экранированы (см. рисунок 3).

#### 1.4.1.1. Способ передачи

Для PROFIBUS принят способ передачи по протоколу RS485, базирующийся на полудуплексной, асинхронной передаче. Данные передаются внутри 11-разрядного кадра (рисунок 4) в NRZ-коде (Non Return to Zero). Значения сигнала (биты) не изменяются во время передачи сигнала.

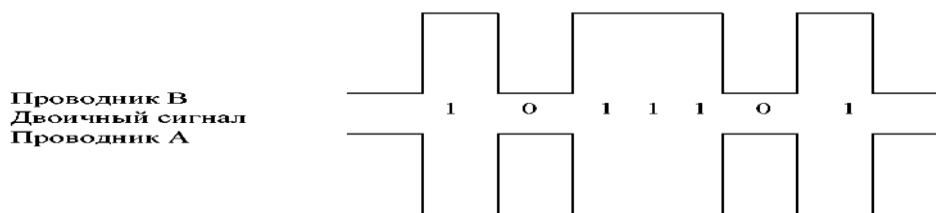
В то время, как передача бинарного значения “1” соответствует положительному значению на проводнике RxD/TxD-P (Receive/Transmit-Data-P), напротив, на проводнике RxD/TxD-N (Receive/Transmit-Data-N) присутствует “0”.





**Рисунок 4 - UART-кадр PROFIBUS**

Состоянию покоя между отдельными телеграммами соответствует двоичный сигнал “1” (рис.5). В литературе часто также оба проводника PROFIBUS обозначают как А-проводник и В-проводник. При этом А-проводник соответствует RxD/TxD-N, а В-проводник -RxD/TxD-P.



**Рисунок 5 - Структура сигнала при передаче IMXX- колом**

### 1.4.2. Канальный уровень (Уровень 2)

На втором уровне реализуются функции управления доступом к шине, защита данных, а также формирование телеграммы и выполнение протокола передачи данных.

Протокол PROFIBUS имеет специальные функции проверки передаваемых данных и может распознавать различные виды ошибок, такие как ошибка протокола, ошибка длины телеграммы, ошибка четности и т. д. В случае возникновения ошибки телеграмма может повторно посылаться до 8 раз. Количество повторов устанавливается шинными параметрами.

По сети PROFIBUS может осуществляться как широковещательная передача данных, так и передача данных «точка-точка».

#### 1.4.2.1. Шинная топология

Различают физическую и логическую топологии сети. Под физической топологией понимается то, как соединяются средства автоматизации между собой. Логическая топология сети определяет, как передается информация от одного узла сети к другому.

При использовании метода передачи RS 485 в сети PROFIBUS используется активная шинная физическая топология. Сеть делится на несколько сегментов с количеством участников не более 32. Если необходимо подключить на одну шину большее количество участников, то используются повторители, которые объединяют несколько сегментов сети. Всего на одной шине может быть до 127 участников.

При использовании оптоволоконной технологии физические топологии сети имеют большое разнообразие. Могут применяться такие структуры, как звезда, кольцо, древовидная, шинная и их комбинации.

Для надежной коммуникации в сети необходимо, чтобы каждый участник сети получал доступ к шине в определенное время, не конфликтуя с другими участниками сети. Логическая шинная топология в сетях PROFIBUS может строиться по трем принципам:

- ведущий – ведомый (master – slave);
- ведущий – ведущий (master – master; кольцевая топология с передачей маркера

– маркерное кольцо);

- комбинация этих двух способов (гибридный способ).

При этом коммуникация «ведущий – ведомый» является централизованной, то есть только один мастер управляет доступом к шине и опрашивает ведомые устройства. Ведомые устройства в этом варианте не могут передавать данные без запроса мастера и не могут самостоятельно получить доступ к шине за некоторыми исключениями.

### 1.4.2.2. Подключение шины

В качестве стандарта для подключения участников к шине в нормах PROFIBUS EN 50170 рекомендуется 9-и штырьковый штекер, который изображен рисунке 6, а расположение контактов в таблице 1. У каждого участника есть такой разъем с контактами, шинный кабель имеет разъем со штырьковыми контактами.

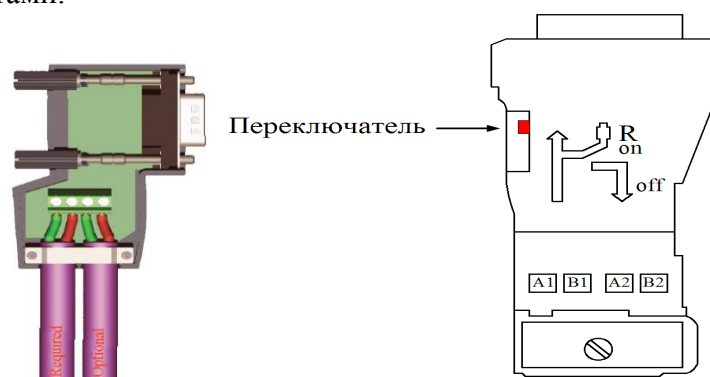


Рисунок 6 - Шинный штекер Sub-D

Таблица 1 Расположение контактов

Вид	Pin-№	Название сигнала	Обозначение
	1	<b>SHIELD</b>	Экран, напр., земля
	2	<b>M24</b>	-24v
	3	<b>RxD/TxD-P</b>	Прием/передача данных, плюс, провод, В
	4	<b>CNTR-P</b>	Сигнал для управления направлением передачи, плюс.
	5	<b>DGND</b>	Данные
	6	<b>VP</b>	Напряжение питания, плюс
	7	<b>P24</b>	+24v
	8	<b>RxD/TxD-N</b>	Прием/передача данных, минус, провод А
	9	<b>CNTR-N</b>	Сигнал для управления направлением передачи, минус.

### 1.4.2.3. Окончание шины

Шинные провода данных с обеих сторон замкнуты на согласованные нагрузки (см. рис.7-8). Благодаря этим сопротивлениям устанавливается безопасный потенциал покоя на проводах шины, когда участники не обмениваются сообщениями (потенциал покоя между телеграммами). Шинные нагрузки имеются почти во всех стандартных разъемах PROFIBUS и могут быть активизированы с помощью переключателей.

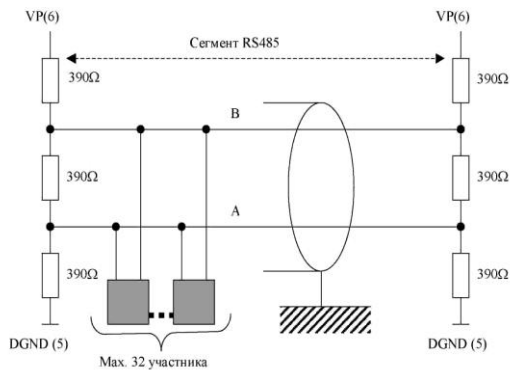


Рисунок 7 Структура шинного сегмента RS485

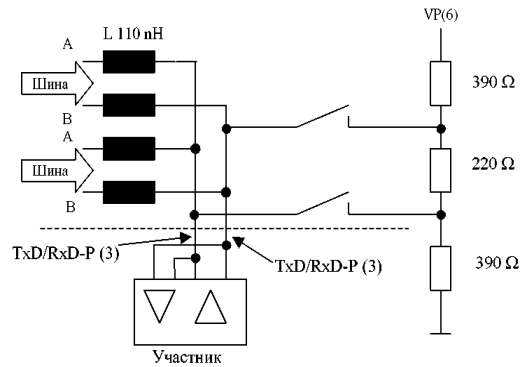


Рисунок 8 - Схема шинного штекера для скорости передачи более 1500 кБит/с

Если используется шина со скоростью передачи более 1500 kBit/s, то нужно на основании потребляемой мощности подключенных участников и отраженной мощности использовать шинный штекер с дополнительной индуктивностью.

### 1.5. Пользовательский интерфейс DP и DP-профили

PROFIBUS-DP применяет уровни 1 и 2. Через пользовательский интерфейс становятся доступными необходимые пользовательские функции, а также системные и аппаратные действия различных типов устройств PROFIBUS-DP.

Протокол PROFIBUS-DP открыто определяет, как передаются между участниками пользовательские данные по шине. Оценка передаваемых по протоколу пользовательских данных не происходит. Благодаря точно установленным параметрам профиля, приборы разных производителей могут совершать обмен.

В настоящее время установлены следующие профили PROFIBUS-DP:

*Профиль для NC/RC*

Профиль описывает, как происходит управление и обслуживание роботов через PROFIBUS-DP. На основании конкретной блок-схемы программы описывается движение и программное управление роботом.

*Профиль для Encoder'а (преобразователя угол-код)*

Профиль описывает присоединение различных Encoder'ов к PROFIBUS-DP. Определены два аппаратных класса основных и дополнительных функций, как, например, масштабирование сигналов и расширенная диагностика.

*Профиль для приводов с изменяемым числом оборотов*

Ведущие производители техники приводов разработали общий PROFIDRIVE-профиль. Профиль устанавливает, как приводы параметрируются и передают заданные и истинные значения. Благодаря этому становится возможным обмен данными приводов различных производителей.

Профиль содержит необходимые установки для вида работы регуляторов числа оборотов и позиционирования. Профиль устанавливает основные функции приводов и дает достаточное свободное пространство для специфических пользовательских расширений. Профиль содержит описание пользовательских функций DP или альтернативных функций FMS.

*Профиль для управления и наблюдения, HMI (Human Machine Interface)*

Профиль устанавливает для приборов обслуживания и наблюдения (HMI) правила подключения этих приборов через PROFIBUS-DP к компонентам автоматизации. Профиль использует для коммуникаций расширенные функции PROFIBUS-DP.

*Профиль для защищенной от ошибок передачи данных через PROFIBUS-DP*

В этом профиле устанавливаются дополнительные механизмы защиты данных для коммуникаций с защищенными от ошибок компонентами, как например, Not-AUS.

### 1.6. Управление доступом к шине в PROFIBUS

К управлению доступом к шине PROFIBUS предъявляются два существенных требования.

С одной стороны для надежных коммуникаций между равноправными приборами автоматизации или РС необходимо, чтобы каждый участник в течение определенного временного окна получал доступ к шине для решения своих коммуникационных задач.

С другой стороны для обмена данными между сложными приборами автоматизации или РС и простой децентрализованной периферией требуется быстрый обмен данными с возможно малыми издержками протокола.

Оба требования удовлетворяются благодаря гибридно построенному управлению доступом к шине, состоящим из:

- децентрализованного обмена маркером (токеном) между активными участниками (Master'ами);
- централизованного обмена Master-Slave для обмена данными между активными и пассивными участниками шины PROFIBUS.

Активный участник, который владеет маркером, берет на себя в данное время функции мастера на шине, чтобы проводить коммуникации с пассивными и активными участниками.

Обмен сообщениями по шине происходит при этом через адресацию участников. Каждому PROFIBUS-участнику назначается однозначный адрес. Адрес назначается из области от 0 до 126. При этом максимальное число участников, находящихся на шине, не превышает 127.

С этим управлением доступом к шине могут быть реализованы следующие конфигурации системы:

- “Чистая” система Master-Master (обмен маркером)
- “Чистая” система Master-Slave (Master-Slave)
- Комбинация обоих методов

Метод доступа к PROFIBUS не зависит от используемой среды передачи, например, медь или оптоволокно.

### 1.6.1. Метод обмена маркером

Активные участники, подключенные к PROFIBUS, упорядочены по возрастанию их адреса в логическое маркерное кольцо (Token-Ring) (рис.9).

Под маркерным кольцом (Token Ring) здесь понимается организационное кольцо из активных участников, в котором маркер (Token) всегда передается от одного участника к следующему. Маркер, а с ним и право на доступ к среде передачи, передается при этом через специальную маркер-телеграмму между активными участниками. Исключение представляет активный участник с наивысшим на шине адресом HSA (Highest Station Address). Он передает маркер исключительно активному участнику с наименьшим шинным адресом, чтобы замкнуть маркерное кольцо.

Время одного обращения маркера через всех активных участников называется *временем обращения маркера*. С помощью устанавливаемого заданного времени обращения маркера Ttr (Time Target Rotation) определяется максимально разрешенное время обращения маркера.



Рисунок 9 – Метод обмена маркером (токеном)

При этом устанавливаются адреса всех имеющихся на шине активных участников и заносятся в LAS (List of Active Station – список активных станций). Для управления маркером при этом

особенно важны адреса предыдущей станции PS (Previous Station), от которой маркер получается, и следующей станции NS (Next Station), которой маркер предназначается. Кроме того, LAS также нужна, чтобы при текущей работе исключать из кольца вышедших из строя или дефектных активных участников и, соответственно, принимать вновь появившихся участников без помех текущему обмену данными по шине.

### 1.6.2. Метод Master-Slave

Если логическое маркерное кольцо состоит только из одного активного и нескольких пассивных участников, то это соответствует “чистой” системе Master-Slave (рис.10).

Метод Master-Slave делает возможным мастеру (активному участнику), который имеет право прямой передачи, опрашивать назначенных ему Slaves (пассивных участников). Мастер при этом имеет возможность принимать сообщения от Slave, и соответственно, передавать.

Типичная стандартная шинная конфигурация PROFIBUS-DP базируется на этом методе управления шиной. Активная станция (DP-Master) обменивается в циклической последовательности данными с пассивными станциями (DP-Slaves).

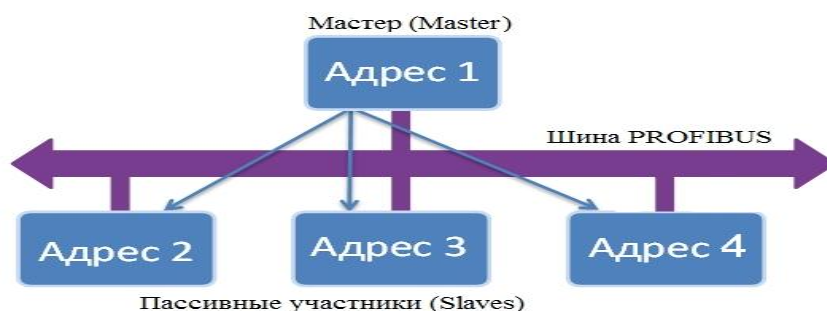


Рисунок 10 - Метод доступа Master-Slave

## 1.7. Цикл PROFIBUS-DP

### 1.7.1. Структура циклов PROFIBUS

На рисунке 11 показывается структура DP-цикла в шинной системе DP с одним мастером.

*Постоянная компонента* DP-циклов составляет при этом часть цикловой телеграммы, состоящей из управления доступом к шине (управление маркером и состоянием участников) и обменом данными (Data\_Exchange) с DP-Slave'ами. Наряду с этим циклическим движением данных есть однако внутри DP-цикла также ряд зависящих от событий, *ациклических телеграмм*.

К этим ациклическим телеграммам относятся:

- Обмен данными во время фазы инициализации DP-Slave
- Диагностические функции DP-Slave'a
- Коммуникации с DP-Master'ом 2-го класса
- Коммуникации с другими мастерами
- Обусловленные уровнем 2 повторения телеграмм при помехах
- Ациклическое движение данных по DP-V1
- On-line функции с PG
- НМІ - функции

В зависимости от этой ациклической телеграммы конкретный DP-цикл может удлиняться.

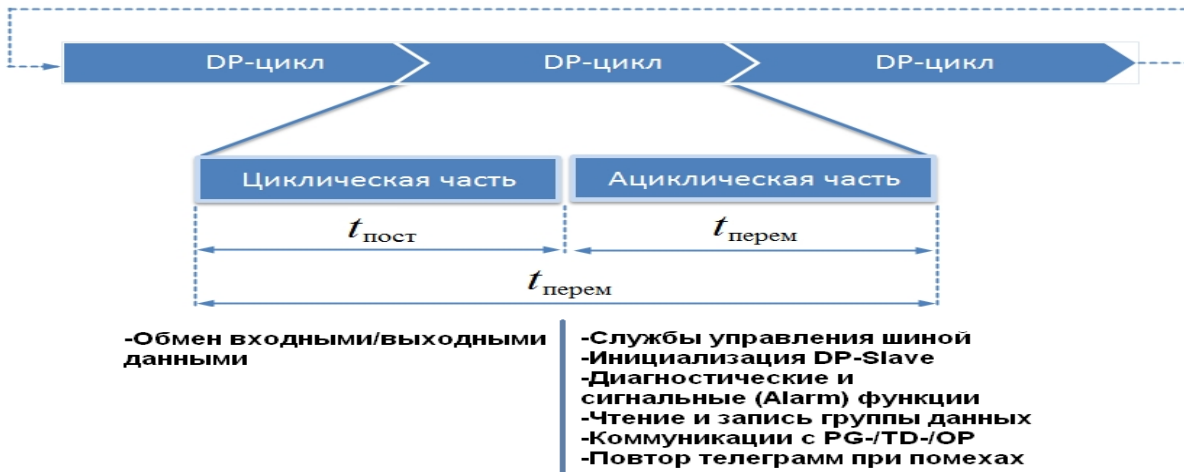


Рисунок 11 - Структура цикла PROFIBUS-DP

Таким образом, шинный цикл всегда состоит из постоянной по времени циклической части и существующей не всегда, зависимой от событий, переменной ациклической части телеграммы.

### 1.7.2. Структура постоянного по времени PROFIBUS-DP цикла

В некоторых случаях при автоматизации выгодно, когда шинный цикл DP по времени остается одинаковым и, таким образом, обмен данными может происходить строго периодически. Это находит применение, например, в области техники приводов для самосинхронизации нескольких приводов.

В отличие от нормального цикла DP, DP-Master'ом предусматривается (резервируется) при постоянном по времени цикле DP для ациклической части коммуникаций определенная часть времени.

Как представлено на рис.12, DP-Master для этого обеспечивает, чтобы эта зарезервированная временная часть была не превышена благодаря тому, что он допускает только определенное число ациклических событий.

Если зарезервированное время не нужно, то DP-Master заполняет паузу “пустым временем”. Благодаря этому гарантируется, что заданное постоянное время будет выдержано с точностью до микросекунды.

Задание времени для постоянного шинного цикла DP осуществляется при создании проекта в STEP 7. Предлагаемое в STEP 7 значение времени определяется, руководствуясь спроектированной конфигурацией установки и учитывая определенные, типичные части ациклических служб. При проектировании постоянного по времени цикла есть возможность изменить предлагаемое STEP 7 значение его длительности.

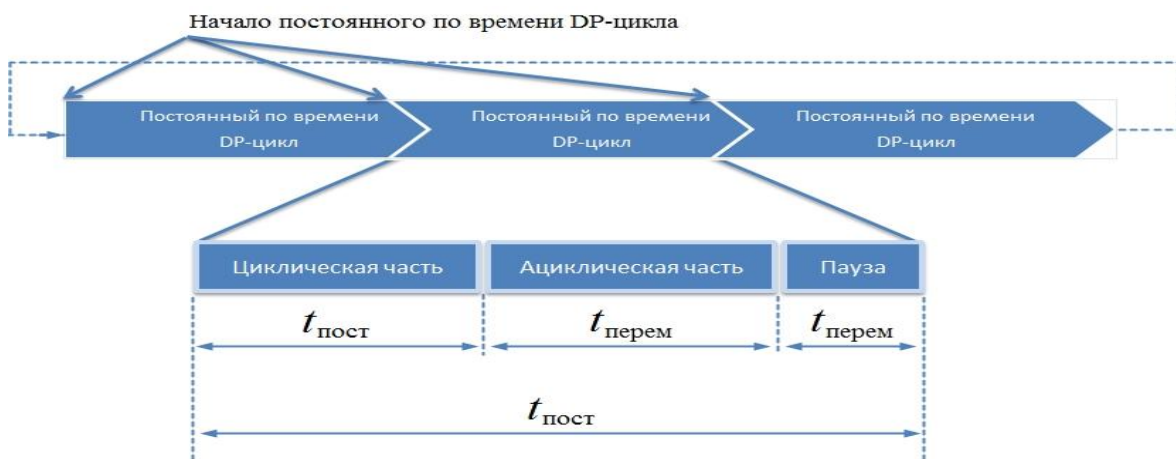


Рисунок 12 - Структура постоянному по времени цикла PROFIBUS-DP

В настоящее время постоянный по времени DP-цикл может быть установлен только в системе с одним мастером.

### 1.7.3. Функциональное расширение DP-V1

Все возрастающие требования к комплексным системам автоматизации приводят к тому, что ведомые устройства помимо своих основных функций должны обеспечивать и дополнительные, к которым относятся:

- диагностические сигналы;
- сигналы от процесса;
- сигнал при удалении блока;
- сигнал при вставке блока;
- сигнал состояния;
- сигнал модернизации.

Все эти функции входят в расширение протокола PROFIBUS-DP DP-V1. Эти сигналы передаются ациклически, при этом для нормальной работы и ведущее, и ведомое устройства в сети должны иметь DP-V1-расширение.

## 2. Взаимодействие с MICROMASTER4 через PROFIBUS-DP

Преобразователи частоты являются исполнительными устройствами и располагаются на нижнем уровне автоматизации технологического процесса. Преобразователи частоты являются ведомыми устройствами в сети и не могут самостоятельно получать доступ к шине, то есть они могут только отвечать на запросы ведущего устройства.

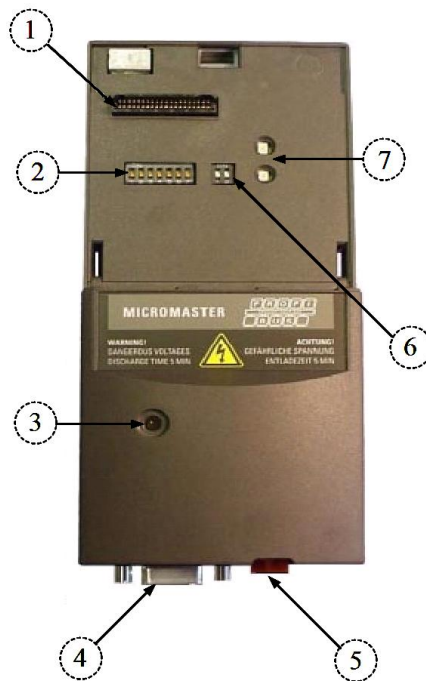
### 2.1. Подключение PROFIBUS-DP к преобразователям частоты и установка адреса

Подключение преобразователя Micromaster 440 к сети PROFIBUS-DP осуществляется за счет дополнительной панели управления, представленная на рис. 13.

Адрес преобразователей в сети PROFIBUS может устанавливаться двумя способами. Первый способ – аппаратный, обозначен цифрой 2 на рис. 13. В этом варианте адрес устанавливается с помощью семи переключателей согласно табл. 2.

Второй – программный вариант установки адреса осуществляется выбором соответствующего значения в параметре **P0918** преобразователя частоты. Аппаратно установленный адрес имеет приоритет, поэтому для установки адреса с помощью параметра все переключатели должны быть установлены в нижнее положение.





**Рисунок 13 - Подключение преобразователя частоты Micromaster 440к сети PROFIBUS-DP:**  
 1 – разъем для устройства управления; 2 – переключатели адресов PROFIBUS;  
 3 – светодиод(ы) рабочих состояний модуля связи; 4 – интерфейс модуля PROFIBUS; 5 – разъем для подачи внешнего питания 24 В; 6 – служебные переключатели; 7 – светодиоды рабочих состояний MICROMASTER4

**Таблица 2 - Выбор адреса с помощью переключателей**

Примеры адресов	Номер и положение переключателя ( $n$ )						
	0	1	2	3	4	5	6
Номер переключателя (бита) - $n$	0	1	2	3	4	5	6
Добавить к значению адреса ( $2^n$ )	1	2	4	8	16	32	64
<b>Пример 1:</b> $3=1+2$	■	■					
<b>Пример 2:</b> $20=4+16$			■		■		

С помощью светодиодов состояния, обозначенных цифрами 3 на рис. 13, можно осуществить диагностику связи преобразователя с сетью. Значения сигналов светодиодов приведены в табл. 3.

**Таблица 3 - Светодиодная индикация модуля связи**

Светодиод	Диагностическая информация
○ – не горит	Нет питания
◐ – красный мигает	Неправильный адрес в сети. Программная либо аппаратная
● – красный горит	Если горит постоянно, это говорит о неисправности модуля связи, либо преобразователя
◑ – желтый мигает	Связь между модулем связи и инвертором установлена, но нет соединения по сети PROFIBUS
● – желтый горит	Связь по сети PROFIBUS установлена, но нет циклического обмена данными
◑ – зеленый мигает	Установлен циклический обмен данными, но неправильное задание или нет задания. Мастер находится в режиме Стоп
● – зеленый горит	Циклический обмен данными

Схема функций связи с приводами MICROMASTER4. реализуемых через PROFIBUS-DP имеет следующий вид (см. рис. 14):

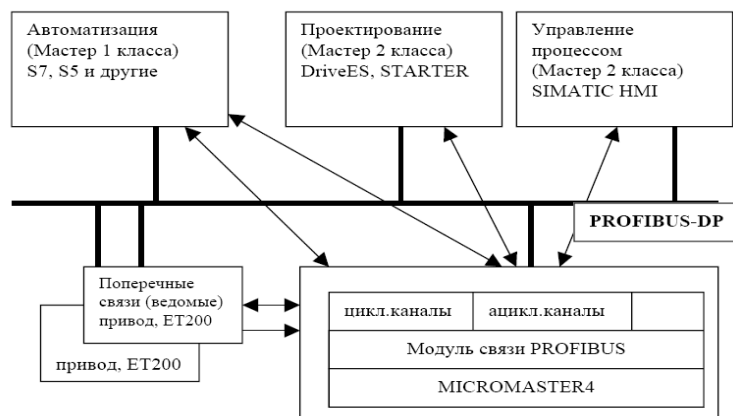


Рисунок 14 - Каналы передачи данных от MICROMASTER4 через PROFIBUS-DP

Управление преобразователем MICROMASTER4 осуществляется по циклическому каналу от PROFIBUS-DP. Дополнительно этим путем может производиться обмен параметрами.

Структура рабочих данных для циклического канала определяется во 2-й версии профиля PROFIDrive и обозначается как параметры данных процесса на объекте (PPO). Профиль PROFIDrive задает структуру рабочих данных для приводов, с помощью которой мастер через циклический обмен данными может обращаться к ведомым приводам.

## 2.2. Структура рабочих данных согласно профилю PROFIDrive 2.0 и 3.0

Структура рабочих данных при циклическом обмене подразделяется на две области, которые могут передаваться в любой телеграмме:

- **Область параметров (PKW)** для считывания/записи значений параметров;
- **Область данных процесса (PZD)**, т.е. управляющие слова и уставки, или же информация о состояниях и фактические значения.

Данные о процессе передаются постоянно. Привод обрабатывает их по наивысшему приоритету и в кратчайшее время. С помощью данных процесса осуществляется управление приводом в рамках системы автоматизации, например включение и отключение, ввод уставок и т.д.

Благодаря области параметров пользователь через систему шин имеет свободный доступ ко всем параметрам, находящимся в преобразователе. Например, он может считывать детальную диагностическую информацию, сообщения о сбоях и т.д.

Таким образом, телеграммы циклической передачи данных имеют следующую базовую структуру рисунок 15:

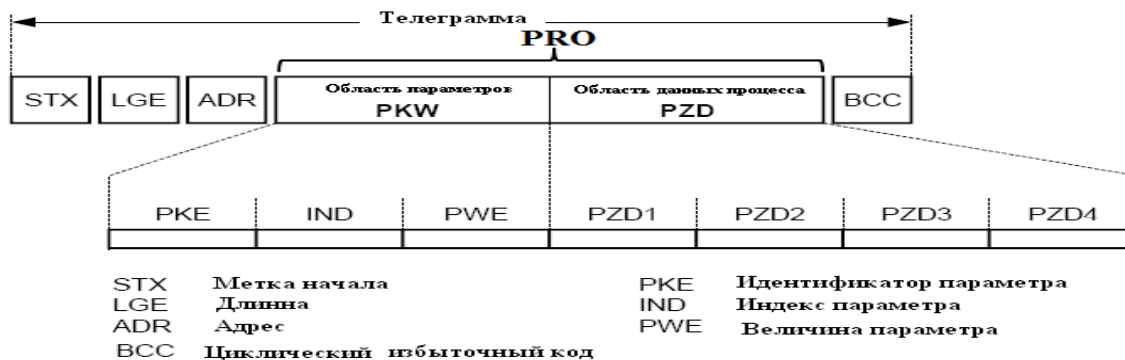


Рисунок 15 - Базовая структура телеграммы циклической передачи данных

Согласно версии 2.0 профиля PROFIDrive определено пять типов PRO (PRO1- PRO5) (см. рисунок 16):

- PRO1- рабочие данные с областью параметров с двумя словами данных процесса;
- PRO2- рабочие данные с областью параметров с шестью словами данных процесса;
- PRO3- рабочие данные без области параметров с двумя словами данных процесса;

- PRO4- рабочие данные без области параметров с шестью словами данных процесса;
- PRO5- рабочие данные с областью параметров с десятью словами данных процесса.

	PKW (Область параметров)				PZD (Область данных процесса)					
	PKE	IND	PWE		PZD1	PZD2	PZD3	PZD4	PZD5	PZD6
	1 слово	2 слово	3 слово	4 слово	1 слово	2 слово	3 слово	4 слово	5 слово	6 слово
PRO1	PKE (идент. параметра)	IND (индекс)	PWE (значение параметра)		STW1 (слово управления 1)	HSW (задание на скорость)	(DP Master → Micromaster)			
	PKE (идент. параметра)	IND (индекс)	PWE (значение параметра)		ZSW1 (слово состояния 1)	HIW (текущая скорость)	(Micromaster → DP Master)			
PRO2	PKE (идент. параметра)	IND (индекс)	PWE (значение параметра)		STW1 (слово управления 1)	HSW (задание на скорость)				
	PKE (идент. параметра)	IND (индекс)	PWE (значение параметра)		ZSW1 (слово состояния 1)	HIW (текущая скорость)		ZSW2 (слово состояния 2)		
PRO3					STW1 (слово управления 1)	HSW (задание на скорость)				
					ZSW1 (слово состояния 1)	HIW (текущая скорость)				
PRO4					STW1 (слово управления 1)	HSW (задание на скорость)				
					ZSW1 (слово состояния 1)	HIW (текущая скорость)		ZSW2 (слово состояния 2)		
PRO5					max 10 PZD (максимум 10 слов)					
					max 10 PZD (максимум 10 слов)					

PKW: величина присвоенного параметра  
PZD: данные процесса  
PKE: присвоение параметров .  
IXD: индекс  
PWE: величина параметра

STW: Управляющее слово 1  
ZSW: Слово состояния 1  
HSW: Главная уставка  
HIW: Главное фактич. значение

**Рисунок 16 -Типы и структура параметров данных процесса на объекте (PRO)**  
**ВНИМАНИЕ!** MICROMASTER4 поддерживает только PRO1 и PRO3 (выделено серым цветом)

Наряду с разделением по типам PRO возможна свободная конфигурация циклических данных. В MICROMASTER4 допускается конфигурирование до 4 слов данных о процессе, в том числе с различным количеством уставок и фактических величин (рис. 16). Области консистенции имеют гибкие границы установки. Область параметров (PKW) можно конфигурировать независимо от количества данных о процессе.

	PKW				PZD									
	PKE	IND	PWE		PZD1	PZD2	PZD3	PZD4	PZD5	PZD6	PZD7	PZD8	PZD9	PZD10
	1 <sup>st</sup> word	2 <sup>nd</sup> word	3 <sup>rd</sup> word	4 <sup>th</sup> word	1 <sup>st</sup> word	2 <sup>nd</sup> word	3 <sup>rd</sup> word	4 <sup>th</sup> word	5 <sup>th</sup> word	6 <sup>th</sup> word	7 <sup>th</sup> word	8 <sup>th</sup> word	9 <sup>th</sup> word	10 <sup>th</sup> word
Max.														
Max.														

**Рисунок 17 - Параметры данных о процессах на объектах (свободная конфигурация)**

Каким типом PRO мастер PROFIBUS-DP будет обращаться к преобразователю задается при наладке системы шин в рамках конфигурации данных для мастера. Выбор соответствующего типа PRO зависит от задач привода в системе автоматизации.

Задание тип PRO осуществляется помощью утилиты HW (GSD - базовые данные аппарата) проекта Step 7. Дополнительно благодаря инжиниринговой системе Drive ES возможно свободное конфигурирование.

### 2.2.1. Слово управления/состояния и сигнал задания на скорость

Профили PRO3 для преобразователей Micromaster 440 являются минимально необходимыми для управления преобразователем по сети PROFIBUS-DP.

Слово управления состоит из шестнадцати бит, каждый из которых отвечает за какую-либо функцию, например пуска, торможения, реверса и др.

Слова управления 1 приведено в табл. 4.

**Таблица 4 - Слово управления 1**

Номер бита	Значение	Функция	Описание
0	1 0	Вкл. Откл. 1 (OFF1)	Переводит привод в состояние «готовность к работе». Откл. 1 – остановка по задатчику интенсивности
1	1 0	Режим работы Откл. 2 (OFF2)	– Откл. 2 – немедленная блокировка импульсов – остановка на
2	1 0	Режим работы Откл. 3 (OFF3)	– Откл. 3 – быстрая остановка – остановка с минимальным
3	1 0	Разрешение работы Запрещение работы	Управление в замкнутой системе и деблокировка импульсов управления Управление в замкнутой системе и блокировка импульсов управления
4	1 0	Режим работы Блокировка ЗИ	– Блокировка задатчика интенсивности (ЗИ) – сигнал на выходе ЗИ устанавливается в ноль.
5	1 0	Деблокировка ЗИ Удержание ЗИ	Нормальная работа задатчика интенсивности «Замораживание» текущей уставки задатчика интенсивности
6	1 0	Деблокировка задания Блокировка задания	Заданное значение подается на вход задатчика интенсивности Значение на входе задатчика интенсивности устанавливается в
7	1 0	Подтверждение ошибки –	Ошибка квитируется по положительному фронту. Преобразователь после этого переходит в состояние запрета на включение
8	1 0	Толчок вправо –	При наличии единицы привод вращается «вправо» с заданной частотой
9	1 0	Толчок влево –	При наличии единицы привод вращается «влево» с заданной частотой
10	1 0	Действительная уставка Уставка недействительна	Мастер передает корректное значение заданной скорости Мастер передает некорректное значение заданной скорости
11	1 0	Инверсия уставки Уставка не инвертируется	Двигатель вращается при положительной уставке влево Двигатель вращается при положительной уставке вправо
12	1 0	– –	Не используется
13	1 0	МОР вверх –	Мотор потенциометр – вверх –
14	1 0	МОР вниз –	Мотор потенциометр – вниз –
15	1 0	Локальное управление	Локальное управление (панель управления, аналоговый вход) Удаленное управление

Несколько примеров наиболее распространенных команд приведены в табл. 5.

**Таблица 5 - Примеры «Слово управления 1»**

Команда	Слово управления в двоичном формате (номера битов):	Слово управления в 16-тиричном формате
	12 8 4 0	
Готовность к работе (стоп)	0000 0100 0111 1110	047E
Запуск вперед	0000 0100 0111 1111	047F
Запуск назад (реверс)	0000 1100 0111 1111	0C7F
Сброс ошибки	0000 0100 1111 1110	04FE

Перед запуском преобразователя от ведущего устройства (мастера) необходимо подготовить

преобразователь к работе. Для этого нужно отправить значение **047E hex** в область слова управления 1. Затем можно подавать команду на включение **047F hex**. После этого, если заданная частота не равна нулю, произойдет запуск привода.

Для сигнала задания на частоту также выделяется 16 бит, как и для слова управления. По умолчанию сигналу задания на частоту в 50 Гц соответствует величина 4000 hex. Эта величина соответствует значению  $2^{14}$ , то есть под номинальное значение частоты выделяется 14 разрядов. Тогда с учетом того, что старший разряд зарезервирован под знак числа, максимальное значение частоты, которое можно задать по сети, составляет приблизительно 100 Гц или 7FFF hex. Однако в преобразователе частоты предусмотрен параметр P2000, в котором можно изменить нормализацию частоты, то есть изменить соотношение 50 Гц – 4000 hex.

Пересчет заданной частоты в 16-тиричный формат проще всего производить через промежуточные значения в десятичном формате.

Для примера несколько значений заданной частоты в 16-тиричном и десятичном форматах приведены в табл. 6.

**Таблица 6 - Примеры значений заданной частоты**

Заданная частота, Гц	Значение заданной частоты в десятичном формате (dec)	Значение заданной частоты в 16-тиричном формате (hex)
50	16384	4000
25	8192	2000
10	3276	CCC
5	1638	666

В принципе «Слова управления 1» и сигнала задания на частоту достаточно, чтобы управлять приводом.

Однако для надежной работы желательно также получать сигналы готовности преобразователя к работе, сигналы ошибок, сигналы предупреждений и др. Это осуществляется посредством слова состояния 1, которое расписано в табл. 7.

**Таблица 7 - Слово состояния 1**

Номер бита	Значение	Функция	Описание
0	1 0	Готовность к включению Готовность отсутствует	Питание преобразователя включено, электроника в исходном состоянии, импульсы управления отключены
1	1 0	Готовность к работе Готовность отсутствует	См. «Слово управления 1», бит 0. Преобразователь включен. При поступлении команды «Работа разрешена» происходит пуск преобразователя Преобразователь не готов к работе (нет команды на включение, неисправность, откл. 2, откл. 3, блокировка)
2	1 0	Работа разрешена Работа запрещена	См. «Слово управления 1», бит 3
3	1 0	Ошибка –	Код ошибки отображается в параметре r0947. После квитирования ошибки происходит блокировка включения
4	1 0	– Команда Откл. 2	– См. «Слово управления 1», бит 1
5	1 0	– Команда Откл. 3	– См. «Слово управления 1», бит 2
6	1 0	Блокировка включения Блокировка отсутствует	Требуется повторное включение через Откл. 1 и затем Вкл. (см. слово управления, бит 0)
7	1 0	Предупреждение –	Выводится в параметре r2110. Привод остается в работе

8	1 0	Допустимая ошибка по скорости Недопустимая ошибка по скорости	Ошибка по скорости находится в допустимых пределах
9	1 0	Требуется управление от мастера Местное управление	Требуется, чтобы мастер взял управление на себя Мастер не имеет приоритета на управление
10	1 0	$f_{max}$ достигнуто $f_{max}$ не достигнуто	Исходная частота больше или равна максимальной частоте
11	1 0	– Достигнуто ограничение по току	Достигнут предельный ток двигателя
12	1 0	– Тормоз двигателя	Сигнал может быть использован для управления тормозом
13	1 0	– Перегрузка двигателя	Данные говорят о перегрузке двигателя
14	1 0	Вращение вправо Вращение влево	
15	1 0	– Перегрузка преобразователя	Например, по току или температуре

Значение текущей частоты вращения двигателя передается в том же формате, что и заданная частота. Значение текущей частоты может не всегда отражать скорость вращения двигателя, а может отображать значение сигнала задания на частоту. Это зависит от выбранной системы управления в преобразователе и наличия либо отсутствия датчика скорости.

Некоторые примеры наиболее распространенных слов состояния 1 приведены в табл. 8.

Таблица 8 - Примеры «Слово состояния 1»

Состояние	Слово состояния в двоичном формате (номера битов):				Слово состояния в 16-тиричном формате
	12	8	4	0	
Готовность к работе (стоп)	1111	1010	0011	0001	FA31
Привод запущен вперед	1111	1011	0011	0100	FB34
Привод запущен назад	1011	1011	0011	0100	BB34
Сигнал ошибки	1111	1010	0011	1000	FA38

### 2.2.2. Механизм PKW для обработки параметров

Механизм PKW в преобразователях используется для чтения или записи параметров по сети PROFIBUS. Механизм может использоваться как при циклической передаче данных, так и при ациклической передаче с помощью расширения DP-V1.

Для просмотра или изменения параметров преобразователя по сети должен быть выбран тип телеграммы, в которой предусмотрена область данных под механизм PKW (см. рис. 15, 16). Например, для преобразователя Micromaster это может быть профиль PRO1

Область данных для обработки параметров всегда состоит из четырех слов и представлена на рис. 18.

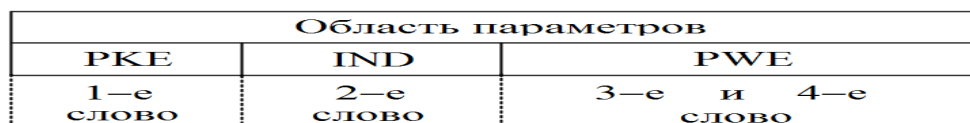
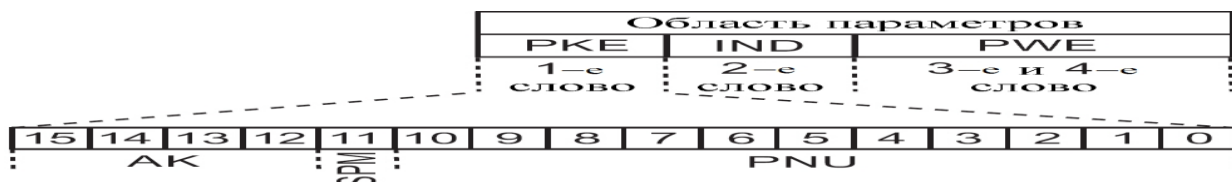


Рисунок 18 - Структура области параметров в телеграмме

Рассмотрим отдельно составные части области параметров (PKW).

Первое слово занимает 16 бит и предназначено для идентификатора параметра (PKE). Структура этой области показана на рис. 19.



**Рисунок 19 - Структура слова идентификатора параметра (PKE)**

Идентификатор параметра (PKE) состоит из следующих областей:

- биты от 0 до 10 (PNU – parameter number) содержат номер параметра, максимум от 0 до 2048 (2<sup>11</sup>), поэтому если номер параметра превышает величину 2000, то дополнительно устанавливается смещение в области (IND), которое будет рассматриваться далее;
- бит 11 (SPM) резервный и его значение всегда равно нулю;
- разряды с 12-го по 15-й (AK) содержат метки задания/ответа.

Для телеграммы с заданием, которая передается от мастера к преобразователю, значение меток задания сведено в табл. 9. В этой же таблице приведены варианты ответов преобразователя на команду мастера для двух случаев.

В табл. 9 горизонтальной линией отделены команды, которые используют память EEPROM. Важно отметить, что циклическое обращение к этой памяти может привести к ее повреждению, поэтому без необходимости не следует пользоваться этими командами.

Для случая успешного выполнения команды приведен код положительного ответа, аналогично приводятся коды отрицательного ответа. Расшифровка кодов ответа приводится в табл. 10.

В табл. 10 горизонтальной линией отделены отрицательные метки ответов 7 и 8.

Когда невозможно выполнить запрос мастера (таб. 9), передается отрицательный ответ (7 – задание невыполнимо, табл. 10) от преобразователя. Причина отрицательного ответа записывается в область PWE. Номера сбоев приведены в табл. 11.

Эти номера позволяют пользователю идентифицировать ошибку и после этого сформировать корректный запрос к параметрам преобразователя.

Второе слово в области обработки параметров (PKW) – это индекс параметра (IND). Структура этого слова зависит от того, как передаются данные, циклически или ациклически. На рис. 20 приведена структура индекса параметра для этих двух видов передачи данных.

Как видно из рис. 20, вид структуры индекса для циклической и ациклической передачи отличается только тем, что субиндекс и индекс страницы меняются местами.

Далее будем подробно рассматривать только циклический обмен данными. В преобразователях предусмотрено несколько наборов параметров, каждый набор имеет свой субиндекс. По средствам сети можно обращаться к различным субиндексам параметров. Субиндекс параметра при циклической передаче данных занимает старшие восемь бит индекса и может принимать значение от 0 до 255 (2<sup>8</sup>).

**Таблица 9 - Метки задания (Мастер → преобразователь)**

Метка задания	Функция	Метка ответа	
		положительная	отрицательная
0	Задание отсутствует	0	7 или 8
1	Запросить величину параметра	1 или 2	7 или 8
2	Изменить параметр (слово)	1	7 или 8
3	Изменить параметр (двойное слово)	2	7 или 8
4	Запросить элемент описания	3	7 или 8
6	Запросить величину параметра (Аггау)	4 или 5	7 или 8
7	Изменить параметр (Аггау, слово)	5	7 или 8
8	Изменить параметр (Аггау, двойное слово)	5	7 или 8
9	Запросить число элементов Аггау	6	7 или 8



11	Изменить параметр (Аггау, двойное слово) и записать в EEPROM	5	7 или 8
12	Изменить параметр (Аггау, слово) и записать в EEPROM	4	7 или 8
13	Изменить параметр (двойное слово) и записать в EEPROM	2	7 или 8
14	Изменить параметр (слово) и записать в EEPROM	1	7 или 8

**Таблица 10 - Метки ответов (Преобразователь → мастер)**

Метка ответа	Функция
0	Отсутствие ответа
1	Перенос параметра (слово)
2	Перенос параметра (двойное слово)
3	Перенос элемента описания
4	Перенос параметра (Аггау, слово)
5	Перенос параметра (Аггау, двойное слово)
6	Перенос количества элементов Аггау
7	Задание невыполнимо (см. табл. 11)
8	Отсутствие приоритета в обслуживании для интерфейса РКВ

**Таблица 11 - Номера сбоя при метке ответа «Задание невыполнимо»**

Номер сбоя	Описание	Причина
0	Недопустимый номер параметра (PNU)	Параметр отсутствует
1	Параметр не изменяется	Параметр только для чтения (r)
2	Выход за нижний/верхний предел	-
3	Неправильный субиндекс	-
4	Отсутствует Аггау	Обращение к простому параметру с заданием Аггау и субиндексом >0
5	Ошибочный тип данных	Перепутаны слово/двойное слово
6	Установка не разрешена (не сбрасывается)	-
7	Элемент описания не изменяется	-
11	Отсутствует приоритет в обслуживании	Описание в преобразователях, как правило, не изменяется
12	Отсутствует ключевое слово	Задание на изменение при отсутствии приоритета в обслуживании (P0927)
17	Задание невыполнимо по рабочему состоянию	-
101	Номер параметра в данный момент деактивирован	Состояние преобразователя в данный момент не позволяет выполнить поставленное задание
102	Ширина канала очень мала	Зависит от состояния преобразователя
104	Недопустимое значение параметра	-
102	Задание не имплементируется	-
104	Выход за модифицированный нижний/верхний предел	Ответ не проходит по каналу коммуникации
106	Значение параметра не может быть изменено из-за отсутствия права доступа	Параметр может иметь только определенные значения
200/201		По маркировке задания 5, 10, 15
204		Минимум/максимум в ходе работы может быть еще более ограничен
		-

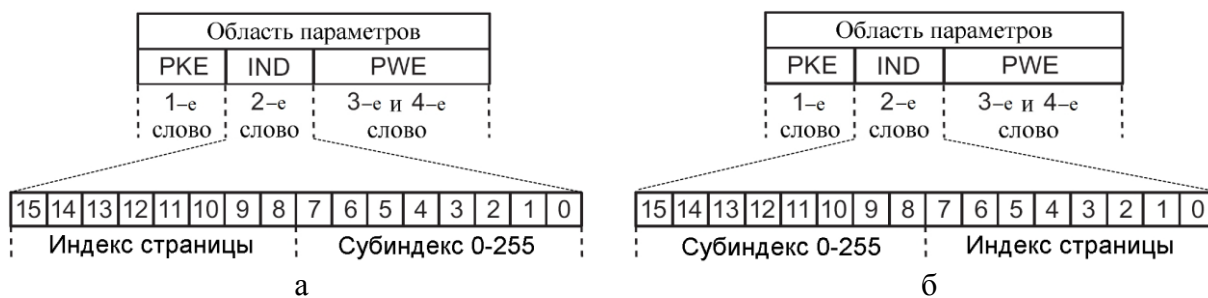


Рисунок 20 - Структура индекса параметра (IND):  
а – ациклическая; б – циклическая

Индекс страницы предназначен для обращения к параметрам, номера которых выше чем 2000. Механизм установки индекса страницы приведен в табл. 12.

Таблица 12 - Установка индекса страницы

Номер параметра	Индекс страницы				Номер бита				16-тиричное значение	+ Номер параметра (PNU)
	7	6	5	4	3	2	1	0		
0000-1999	0	0	0	0	0	0	0	0	0	0-7CF
2000-3999	1	0	0	0	0	0	0	0	0x80	0-7CF
4000-5999	0	0	0	1	0	0	0	0	0x10	0-7CF
6000-7999	1	0	0	1	0	0	0	0	0x90	0-7CF
8000-9999	0	0	1	0	0	0	0	0	0x20	0-7CF
...	...	...	...	...	...	...	0	0	...	0-7CF

Для установки индекса страницы используются только младшие 6 бит. Биты 6 и 7 всегда равны нулю.

**Важно** - во второй версии PROFIdrive для преобразователей Micromaster используются только два верхних значения индекса страницы 0 и 1 (бит 7), которые перекрывают весь диапазон номеров параметров для этого преобразователя.

Рассмотрим в качестве примера номер параметра P2020. Для выбора этого параметра нужно в область номера параметра PNU записать значение 20 (14 hex) и выбрать вторую страницу, установив значение 1 согласно табл. 12.

Значение параметра задается в области PWE, которая всегда занимает два слова (32 бита). 32-битное значение параметра передается через старшее PWE1 и младшее PWE2 слово.

А для передачи 16-битного значения параметра используется только младшее слово PWE2 (рис.21).

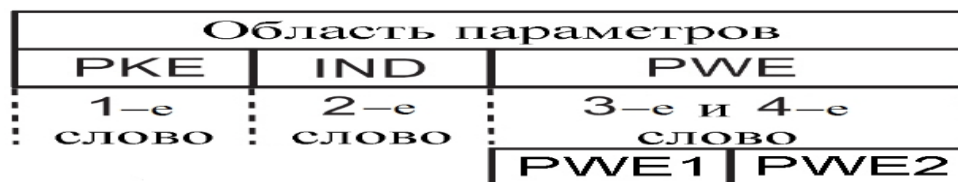


Рисунок 21 - Область значений параметров

**При работе с параметрами соблюдаются следующие правила:**

- обращение или ответ всегда относятся только к одному параметру;
- мастер должен повторять задание до тех пор, пока не будет получен нужный ответ. Ответ может распознаваться обработкой метки ответа или другими способами;
- задание должно полностью передаваться одной телеграммой. Разбивать телеграммы на части не разрешается. Это же самое относится и к ответу. Это в первую очередь относится к ациклической передаче данных;

- в ответных телеграммах, которые содержат значения параметров, привод с каждым повтором ответа посылает текущие значения;
- если в циклическом режиме информация от интерфейса PKW не требуется (значение имеют только данные PZD), то задание должно быть сформулировано как «отсутствие задания».

Рассмотрим некоторые примеры обращения к параметрам преобразователя, приведенные в табл. 13.

Часть параметров в преобразователе представлена в формате с плавающей точкой IEEE 754. Для изменения такого параметра нужно сначала провести преобразование из десятичного формата в 16-тиричный.

Перед отправкой команды на изменение параметра желательно сначала отправить запрос, чтобы узнать тип параметра (16- или 32- разрядный) и проверить номер параметра и его значение.

**Таблица 13 - Примеры использования механизма PKW по сети PROFIBUS**

Описание		PKW (16-тиричный формат)				Примечание
		PKE	IND	PWE		
Чтение P0700	Запрос	12BC	0000	0000	0000	P0700 - выбор источника команд (2 – дискретные входы)
	Ответ	12BC	0000	0000	0002	
Чтение P1000	Запрос	13EB	0000	0000	0000	P1000 – выбор источника задания на частоту (2 – аналоговый вход)
	Ответ	13EB	0000	0000	0002	
Чтение P1082	Запрос	143A	0000	0000	0000	P082 – максимальная частота двигателя (50 Гц – 42480000 IEEE float point)
	Ответ	243A	0000	4248	0000	
Чтение P2010.1	Запрос	100A	0180	0000	0000	P2010 индекс 1 – максимальная скорость передачи USS. (6 – 9600 кбит/с)
	Ответ	100A	0180	0000	0006	
Запись P1082	Запрос	343A	0000	4220	0000	P082 – максимальная частота двигателя (40 Гц – 44200000 IEEE float point)
	Ответ	243A	0000	4220	0000	
То же, но в ответ ошибка	Запрос	143A	0000	0000	0000	Ошибка – привод запущен и не может изменить параметр (код 17 табл. 12)
	Ответ	743A	0000	0000	0011	

### 2.2.3. Параметрирование преобразователя для работы в сети PROFIBUS-DP

Для работы преобразователя частоты от сети PROFIBUS нужно задать в преобразователе соответствующие параметры. Список параметров, которые относятся к настройке соединения и управлению приводом по сети PROFIBUS, приведен в табл. 14.

**Таблица 14 - Параметры для настройки соединения и управления преобразователем по сети**

Параметр	Описание	Значение по умолчанию
<b>P0918</b>	Адрес PROFIBUS (0 – 255)	3
<b>P0719</b>	Приоритетность в управлении данными процесса	0
<b>P0700</b>	Быстрый выбор источника управляющих команд	2
<b>P1000</b>	Быстрый выбор источника задания частоты	2
<b>r2050</b>	Источник данных процесса (BICO) (от мастера к преобразователю)	0
<b>P2051</b>	Фактические значения процесса (BICO) (от преобразователя к мастеру)	K 52
<b>P2041</b>	Функции модуля связи	0
<b>P2040</b>	Время отсутствия телеграммы с данными	20
<b>P0927</b>	Источник изменения для параметров	1111
<b>r2054</b>	Диагностика модуля связи	0

Рассмотрим более подробно назначение каждого из параметров.

Параметр P0918 задает адрес преобразователя частоты в сети PROFIBUS-DP. Может принимать значения от 1 до 255. Адрес может также устанавливаться аппаратно с помощью

переключателей (см. рис. 11). Аппаратный адрес имеет более высокий приоритет и изменение параметра P0918 становится недоступным.

Значение параметра P0719=66 передает приоритет управления данными процесса (PZD) на сеть PROFIBUS. Однако слово управления/состояния и заданная и текущая частота передаются независимо от значения этого параметра, поэтому его установка необязательна. При использовании технологии BICO значение параметра должно устанавливаться в ноль.

Значения параметров P0700=6 и P1000=6 полностью передают управление преобразователем на сеть.

Параметры r2050 и P2051 используют технологию BICO для более гибкой настройки телеграммы. Оба этих параметра имеют 8 индексов, что соответствует максимальной длине телеграммы (за исключением PRO5, которая настраивается с помощью DriveES Simatic). Через параметры r2050 и P2051 по средствам бинкетов и коннекторов могут выбираться данные, которые принимаются и отправляются по сети. Приведенная ниже таблица показывает, как используются параметры r2050 и P2051 для передачи данных процесса.

**Таблица 15 - Передача данных процесса с помощью параметров r2050 и P2051**

Телеграмма	PZD1 STW/ZSW	PZD2 HSW/HiW	PZD3	PZD4
Ссылки для источников задания и управляющих команд (мастер → преобразователь)	r2050.00	r2050.01	r2050.02	r2050.03
Ссылки для действительных величин (преобразователь → мастер)	P2051.00	P2051.01	P2051.02	P2051.03

Для большинства телеграмм первые два индекса параметров r2050 и P2051 заняты по умолчанию под «Слово управления/состояния» и «Заданное/текущее значение скорости».

Остальные параметры не представляют особого интереса и с их назначением можно ознакомиться в соответствующей документации [1]

### 3. PROFIBUS-DP в системе SIMATIC S7

PROFIBUS – составная часть системы SIMATIC S7. Децентрализованная периферия (DP) обрабатывается благодаря связи через проект STEP 7, как центральная периферия. Поведение системы при выходе из строя, диагностике и alarm'ах от SIMATIC S7 DP-Slave'ов также соответствует поведению центральной периферии.

Через встроенный или устанавливаемый интерфейс PROFIBUS-DP можно также подключать к системе автоматизации полевые приборы со сложными техническими функциями. Свойства PROFIBUS, определенные на уровнях 1 и 2 и прозрачность внутренних системных коммуникационных возможностей (S7-функции) позволяют эксплуатировать в системах SIMATIC S7 PROFIBUS-DP программаторы (PG), PC, а также приборы обслуживания и наблюдения.

#### 3.1. DP-интерфейсы в системах SIMATIC S7

В системах SIMATIC S7-300 и SIMATIC S7-400 различают два варианта интерфейсов PROFIBUS-DP:

- *Встроенные в CPU интерфейсы* (CPU 315-2, CPU 316-2, CPU 318-2, CPU 412-1, CPU 412-2, CPU 413-2, CPU 414-2, CPU 414-3, CPU 416-2, CPU 416-3, CPU 417-4)
- *Интерфейс, образованный с помощью подключения* в PLC IM (интерфейсного модуля) или CP (коммуникационного процессора) (IM467, IM467-FO, CP443-5 (Extended), CP342-5)

В зависимости от рабочих характеристик CPU различают также рабочие характеристики DP-интерфейса. За исключением CP342-5 децентрализованная периферия обрабатывается через DP-интерфейс так же, как центральная. DP-интерфейс CP342-5 работает отдельно от CPU. Обмен пользовательскими данными происходит через вызов специальных функций (FC) из программы пользователя.

### **3.1.1. Свойства запуска интерфейса DP-Master в SIMATIC S7**

При децентрализованной структуре установки часто по техническим и топологическим причинам не возможно подключить все электрические машины или части установки. На практике это при известных обстоятельствах означает, что при запуске DP-Master'a еще не все запрограммированные DP-Slave'ы имеются в наличии. Перед началом циклической работы после включения питания во время фазы запуска (Startup) DP-Master должен параметризовать и конфигурировать все назначенные ему Slave'ы. Для систем S7-300 и S7-400 можно установить максимальное время ожидания сообщения о готовности всех DP-Slave'ов с помощью параметра *“Finished” Message by Means of Modules*. Область устанавливаемых значений лежит между 1ms и 65000 ms. По умолчанию установлено значение 65000 ms. По истечению этого времени CPU переходит в состояние STOP или RUN в зависимости от установки параметра *“Startup at Present Configuration Not Equal to Actual Configuration”* (*Запуск при несовпадении заданной и действительной конфигураций*).

### **3.1.2. Выход из строя станции DP-Slave**

Выход из строя DP-Slave'ов, например, из-за сбоя источника питания, обрыва шины или ее дефекта, сообщается операционной системой CPU через вызов организационного блока OB86 (выход из строя носителя модулей, DP-сети или DP-Slave). OB86 вызывается операционной системой, как при приходящем, так и при уходящем событии. Если OB86 не запрограммирован, то при выходе из строя DP-сети или DP-Slave'a CPU переходит в состояние STOP. Таким образом, при выходе из строя децентрализованной периферии система SIMATIC S7 ведет себя так же, как при выходе из строя центральной периферии.

### **3.1.3. Сигнал (Alarm) при удалении/вставке модуля**

Модули, расположенные в центральной корзине и спроектированные в системе SIMATIC S7, периодически опрашиваются центральным процессором и при их удалении или вставке CPU получает соответствующий сигнал. SIMATIC DPS7-Slave'ы и DPV1-Slave'ы также могут отслеживать эти события и при их наступлении сообщать о них DP-Master'у. Благодаря этому в CPU стартует блок OB83, при этом он будет при удалении стартовать как приходящее событие, а при вставке – как уходящее. При вставке модуля в спроектированный слот в состоянии работы (RUN) операционная система проверяет, соответствует ли тип вставленного модуля спроектированному типу. Затем стартует OB83 и при совпадении типов спроектированного и вставленного модулей происходит параметрирование (назначение параметров) вставленного модуля. Если OB83 отсутствует в CPU, то при появлении сигнала удаления/вставки центральный процессор переходит в состояние STOP.

### **3.1.4. Диагностические сигналы от станций DP-Slave**

Модули с диагностическими способностями в области децентрализованной периферии в состоянии сообщать о событиях с помощью диагностических прерываний как, например, частичный выход из строя станции, обрыв провода у сигнального модуля, короткое замыкание/перегрузка периферийного канала или выход из строя источника питания. При приходящих или уходящих диагностических сигналах операционная система CPU вызывает организационный блок OB82 для обработки диагностических сигналов. Если OB82 не запрограммирован, CPU переходит в состояние STOP. Возможные диагностические события и структуры их сообщений в зависимости от сложности DP-Slave'ов частично описаны в EN 50170. Внутри DP-Slave SIMATIC S7 возможные диагностические события согласованы с системной диагностикой SIMATIC S7.

### **3.1.5. Сигналы от процесса у станций DP-Slave**

DP-Slave SIMATIC S7, способные генерировать сигналы от процесса, могут сигнализировать через шину CPU DP-Master'у о событиях в процессе, как, например, выход за пределы верхней или нижней границы значения аналогового сигнала. Для обработки сигналов от прцесса в системе SIMATIC S7 зарезервированы организационные блоки OB40 ... OB47, которые вызываются операционной системой в случае поступления сигнала. Таким образом, обработка возбуждаемых

сигналов от процесса в системе SIMATIC S7 идентично, как у децентрализованной, так и у центральной периферии. Следует, однако, принять во внимание, что время реакции на сигналы от процесса, возбуждаемые децентрализованной периферией по сравнению с сигналами от процесса, возбуждаемыми центральной периферией, обусловлено временем прохождения телеграммы по шине и последующей обработкой сигнала в DP-Master'е и поэтому больше.

### 3.1.6. Сигнал состояния (Statusalarm) от DP-Slave'a

DPV1-Slave'ы могут вызывать сигнал состояния. Если модуль DPV1-Slave меняет свое рабочее состояние, например RUN на STOP, то об этой смене состояния может быть сообщено DP-Master'у с помощью сигнала состояния. Точное событие, которое вызывает сигнал состояния, устанавливается изготовителем и может быть взято из документации DPV1-Slave'a. Благодаря сигналу состояния операционная система CPU вызывает организационный блок OB55. Если этот блок не запрограммирован, то CPU не смотря на это остается в состоянии RUN. OB55 имеется только у S7-CPU, поддерживающих DPV1.

## 4. Краткая информация о программном обеспечении STEP 7

### 4.1. Общие сведения о программном обеспечении STEP 7

Программный пакет STEP 7 – базовое программное обеспечение для программирования и проектирования систем SIMATIC S7. Базовый пакет STEP 7 состоит из различных приложений (рис. 21).



Рисунок 22- Структура программного обеспечения STEP 7

Они применяются для:

- Конфигурирования и параметрирования аппаратуры
- Конфигурирования сетей и соединений
- Загрузки и тестирования пользовательской программы

Благодаря ряду дополнительных пакетов, например, таких, как языки программирования SCL, S7GRAPH или HiGraph, базовый пакет STEP 7 расширяется для соответствующих приложений. С помощью центрального инструмента SIMATIC Manager все необходимые приложения могут быть вызваны в графическом виде. Все данные и установки для системы автоматизации структурированы внутри проекта и представлены в виде объектов. Пакет STEP 7 снабжен обширной online-помощью вплоть до контекстной помощи для указанного контейнера, объекта и появляющегося сообщения об ошибке.

### 4.2. Основы STEP 7

#### 4.2.1. Структура программы

В полную программу процесса входят

**Операционная система:** содержит общую часть всех инструкций и соглашений для реализации внутренних функций (например, сохранение данных при сбросе напряжения питания, управление реакцией пользователя при прерывании и т. д.). Она расположена на так называемом EPROMe (Erasable

Programmable Read Only Memory) и является фиксированной составной частью процессора. Как пользователь, вы не имеете возможности обращаться к операционной системе.

**Программа пользователя:** содержит набор всех написанных пользователем инструкций и соглашений для обработки сигналов, с помощью которых производится управление установкой (процессом). Программа пользователя распределяется на блоки. Деление программы на блоки значительно проясняет структуру программы и подчеркивает программно-технические связи отдельных частей установки.

**Блоком** называется часть программы пользователя, ограниченная функционально и структурно или по целям использования.

Различают блоки, которые содержат

- инструкции для обработки сигналов;
- блоки, содержащие данные (блоки данных).

Блоки идентифицируются:

- типом блока (OB, PB, SB, FB, FX, DB, DX);
- номером блока (число от 0 до 255).

### Инструкции языка STEP.

Инструкция языка STEP является наименьшей самостоятельной единицей программы пользователя. Она является предписанием для работы процессора. Инструкция может быть представлена в виде

• **Список команд (инструкций) - STL.** Представляет собой список команд подобно обычному языку Ассемблера.

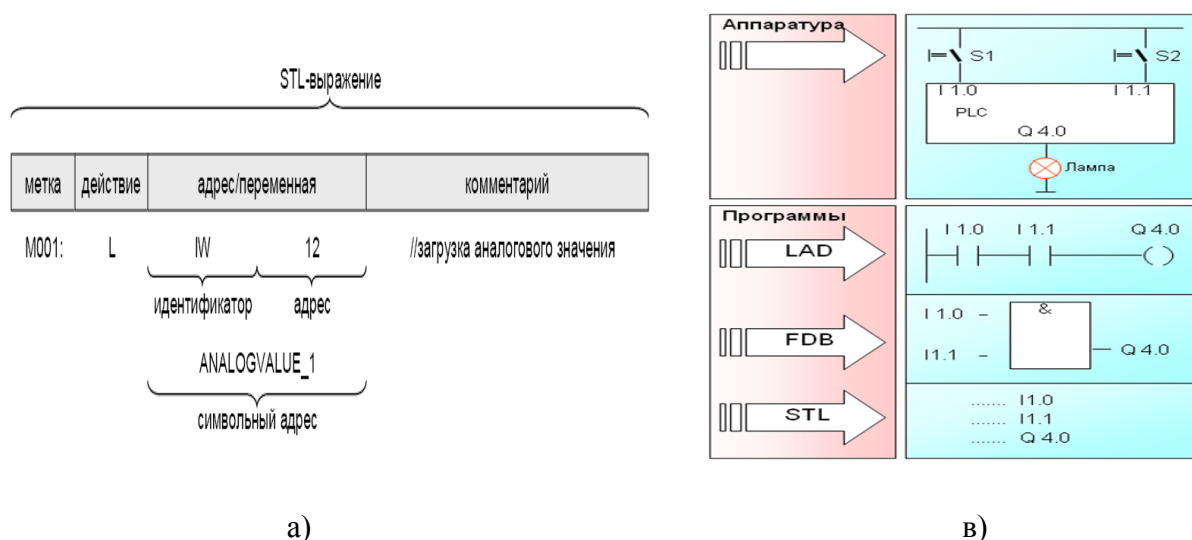
• **Контактный план - LAD.** Управляющая программа записывается при помощи изображений элементов релейных контактных схем.

• **Функциональный план – FUP.** Для отображения программы используются схемы логических элементов.

STL-программа состоит из ряда отдельных выражений (statement). Выражение - это наименьшая самостоятельная единица пользовательской программы. Выражение содержит описание работы для CPU. На рисунке 22а показана общая структура STL-выражения.

Инструкция языка STEP состоит из операции и операнда.

Операнд может быть представлен абсолютно или символически (через список соответствия).



а)

в)

Рисунок 23 - Инструкции языка STEP 7  
а) структура STL-выражения; в) виды представления инструкций языка STEP 7

### Типы блоков



Таблица 1 содержит перечень и краткое описание программных блоков для контроллеров SIMATIC S7, а также блоков данных.

В контроллерах SIMATIC S7 существует несколько способов обработки управляющей программы:

**1. Циклическая обработка.** Состоит из повторных (периодически повторяющихся) обработок управляющей программы, которая начинается с вызова организационного блока OB1. В начале цикла обработки программы ОС заполняет область отображения входов, сбрасывает таймер контроля длительности цикла, после этого вызывает для обработки блок OB1. В конце цикла обработки ОС переписывает в выходные модули значения из области отображения выходов, после чего начинается следующий цикл обработки. В блоке OB1 можно вызывать функции и функциональные блоки. После обработки вызванного блока управление передается блоку, из которого был произведен вызов данного блока.

**Таблица 16-Программные блоки и блоки данных**

Блок	Выполняемые функции
OB	Организационные блоки. Предназначены для: организации циклического выполнения программы пользователя (OB1), обработки событий: <ul style="list-style-type: none"> <li>• включения питания (OB100),</li> <li>• циклического прерывания (OB30-OB38),</li> <li>• прерывания по дате и времени (OB10-OB17),</li> <li>• прерывания по задержке времени (OB20-OB23),</li> <li>• возникновения ошибки (OB40-OB47, OB80-OB87, OB121, OB122).</li> </ul> Блоки вызываются <i>автоматически</i> операционной системой контроллера в случае возникновения того или иного события. <sup>1</sup>
FC	Функция. Может быть вызвана из любого блока. Допускается передача параметров в функцию и обратно. Функция может также иметь локальные переменные, которые теряются при выходе из блока.
FB	Функциональный блок. Также может быть вызван из любого блока и может иметь формальные и локальные параметры. Особенностью FB является наличие переменных типа STAT, которые сохраняют свое значение при выходе из блока. Поэтому функциональный блок имеет один или несколько связанных с ним блоков данных (т.наз.
SFC	Системная функция. Это функция, уже имеющаяся в ОС CPU. Предназначена для выполнения определенных стандартных действий.
SFB	Системный функциональный блок. Аналогичен FB, но, как и SFC. уже имеется в составе ОС контроллера. <sup>2</sup>
DB	Блок данных программы пользователя. Предназначен для долговременного хранения информации.
DI	Блок данных функционального блока. Используется для хранения значений переменных функционального блока. Отличается от DB наличием жесткой структуры, определяемой связанным с ним функциональным блоком.

**2. Циклическая обработка.** Состоит из повторных (периодически повторяющихся) обработок управляющей программы, которая начинается с вызова организационного блока OB1 (рис.23). В начале цикла обработки программы ОС заполняет область отображения входов, сбрасывает таймер контроля длительности цикла, после этого вызывает для обработки блок OB1. В конце цикла обработки ОС переписывает в выходные модули значения из области отображения выходов, после чего начинается следующий цикл обработки. В блоке OB1 можно вызывать функции и функциональные блоки. После обработки вызванного блока управление передается блоку, из которого был произведен вызов данного блока.

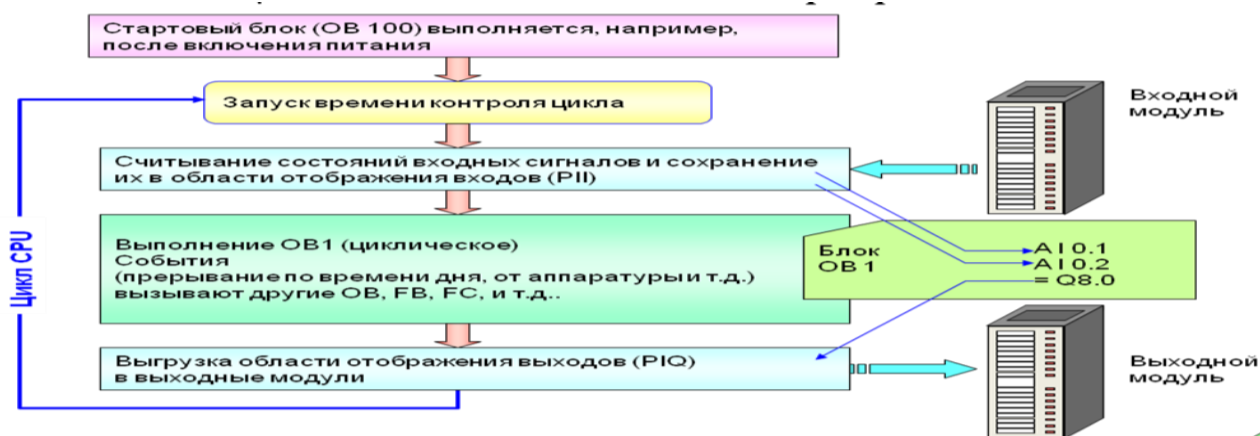


Рисунок 24 - Циклическое выполнение программы пользователя

3. **Циклические прерывания.** При управлении ТП всегда существуют программы, которые должны обрабатываться через одинаковые, заранее заданные, промежутки времени. Для этих целей в контроллерах SIMATIC S7 существуют блоки обработки циклических прерываний. Промежуток времени, через который должен вызываться данный блок, задается программистом.

4. **Прерывания по дате и времени.** Существуют программы, которые должны выполняться один раз в определенный день и час или выполняться периодически, начиная с определенной даты и времени. Для этих целей в контроллерах SIMATIC S7 можно запрограммировать блоки прерываний по дате и времени.

5. **Прерывания по задержке времени.** Такие блоки вызываются по истечении определенного времени после возникновения какого-либо события.

6. **Обработка включения питания.** Часто при включении питания необходимо выполнить какие-либо однократные действия: первичную установку, инициализацию и т.д. Для этих целей предусмотрены блоки обработки включения питания.

7. **Обработка ошибок.** Такие блоки выполняются в случае возникновения аппаратных или программных ошибок.

Организационные блоки обрабатываются циклически. Период и приоритет обработки определяется номером организационного блока (см. таблицу 2).

Таблица 17 - Период и приоритет обработки организационных блоков

Организационный блок	Условия вызова	Приоритет	
		по умолчанию	изменения
OB1	Вызывается операционной системой	1	Нет
OB10 – OB17	В заданное время	2	2 ... 24
OB20 – OB23	По истечении времени	3 ... 6	2 ... 24
OB30 – OB38	Через заданный интервал	7 ... 15	2 ... 24
OB40 – OB47	От входов и выходов	16 ... 23	2 ... 24
OB60	Мультипроцессорное	25	Нет
OB70, OB72, OB73	Ошибки резервирования	25, 28, 25	2 ... 28
OB80 – OB85	Асинхронные ошибки	26	2 ... 26
OB90	Фоновая обработка	29	Нет
OB100	При запуске	27	Нет
OB121, OB122	Ошибки выполнения программы	Приоритет блока, вызвавшего программу	

## Типы данных

Контроллеры SIMATIC S7 могут работать со следующими типами данных (рис. 25):

## 1. Элементарные типы данных (до 32 бит)

а. **Битовые типы данных** представлены следующими типами:

### Бит (BOOL)

Бит - это единица, соответствующая одному двоичному разряду. Два возможных значения бита обозначаются "0" (FALSE) и "1" (TRUE).

### Байт (BYTE)

Байт состоит из 8 бит, которым соответствуют битовые адреса от 0 до 7 (справа налево). Старшим является бит с большим адресом. Байт могут образовывать только те биты, адрес младшего из которых кратен 8, например: 0, 8, 16 и т.д. В контроллерах Simatic S7 байт может интерпретироваться как просто байт (набор бит) или как ASCII-символ.

### Слово (WORD)

Слово - это следующая после байта по величине единица, ее длина 16 бит. Любые два соседних байта можно объединить в слово, старшим будет являться байт с меньшим адресом. Адрес слова - это адрес байта с меньшим адресом. В контроллерах Simatic S7 слово может интерпретироваться как просто слово (набор бит), целое число со знаком, дата, время и т.д.

### Двойное слово (DWORD)

Любые два соседних слова можно объединить в двойное слово, его длина - 32 бита или 4 байта. Старшим словом (байтом) является слово (байт) с меньшим адресом. Адрес двойного слова - это адрес байта с меньшим адресом. В контроллерах Simatic S7 двойное слово можно интерпретировать как просто двойное слово, длинное целое число со знаком, вещественное число в формате IEEE и т.д.

Таблица 18 - Представление битовых типов данных

Представление	Байт (B)	Слово (W)	Двойное слово (D)
Целое без знака	от 0 до 255 от 0 до FF	от 0 до 65 535 от 0 до FFFF	от 0 до 4 294 967 295 от 0 до FFFF FFFF
Целое со знаком	от -128 до +127 от 80 до 7F	от -32 768 до +32 767 от 8000 до 7FFF	от -2 147 483 648 до +2 147 483 647 от 8000 0000 до 7FFF FFFF
Вещественное IEEE 32-битовое плавающей точкой	Неприменимо	Неприменимо	от +1.175495E-38 до +3.402823E+38 (положительное) от -1.175495E-38 до -3.402823E+38 (отрицательное)

### Char (литера)

Переменная типа CHAR (character, литера) занимает один байт. Тип данных CHAR представляет одну литеру в ASCII-формате, например, 'A'.

Работая с этим типом данных, вы можете использовать любую печатную литеру в апострофах.

б. **Математические типы данных** представлены следующими типами:

### INT (целое число)

Переменная типа INT (integer) хранится как целое число (16-битное число с фиксированной запятой или десятичной точкой). Тип данных INT не имеет специального идентификатора.

Целочисленная переменная занимает одно машинное слово. Сигнальные состояния битов с 0-го по 14-ый представляют цифровые разряды (позиции) числа. Сигнальное состояние 15-го бита представляет знак (sign, S).

### DINT (двойное целое число)

Переменная типа DINT хранится как целое число (32-битное число с фиксированной запятой). Целое сохраняется в DINT-переменной, когда оно превышает 32 767 или меньше -32 768, или когда число предваряется идентификатором типа L#.

Под переменную типа DINT отводится двойное слово. Сигнальные состояния битов с 0-го по 30-ый представляют цифровые позиции числа. Знак хранится в 31-м бите.

### REAL (вещественный)

Переменная типа REAL представляет дробь и хранится как 32-битное число с плавающей запятой (десятичной точкой). Целое сохраняется как переменная типа REAL при добавлении десятичной точки.

В экспоненциальном представлении вы можете предварить «e» или «E» целым числом или дробью из семи соответствующих чисел и знака. Цифры, которые расположены за «e» или «E» представляют экспоненту по базе 10. STEP 7 производит преобразование REAL-переменной во внутренне представление числа с плавающей точкой.

с. **Временные типы данных** представлены следующими типами:

#### **S5TIME**

Переменная типа S5TIME используется в базовых языках STL, LAD и FBD для установки таймеров системы SIMATIC. Она занимает одно 16-битное слово с 1 ÷ 3 декадами.

Время устанавливается в часах (hours), минутах (minutes), секундах (seconds) и миллисекундах (milliseconds).

#### **DATE (Дата)**

Переменная типа DATE хранится в машинном слове как число с фиксированной точкой без знака. Содержимое переменной соответствует количеству дней, начиная с 01.01.1990. Ее представление показывает год, месяц и день, разделенные дефисом.

#### **TIME (Время)**

Переменная типа TIME резервирует одно двойное слово. Ее представление содержит информацию о днях (d), часах (h), минутах (m), секундах (s) и миллисекундах (ms), отдельные элементы этих данных могут быть опущены. Содержимое переменной интерпретируется в миллисекундах (ms) и хранится как 32-битное число с фиксированной точкой со знаком.

#### **TIME\_OF\_DAY (Время суток)**

Переменная типа данных TTMEOFDAY резервирует для себя одно двойное слово. Она содержит количество миллисекунд с начала суток (со времени 00:00) в виде числа с фиксированной точкой без знака. Ее представление содержит информацию о часах, минутах и секундах, разделенных двоеточием. Миллисекунды, которые следуют за секундами, отделены от них десятичной точкой. Миллисекунды могут отсутствовать.

## **2. Сложные типы данных (более чем 32 бита)**

STEP 7 определяет следующие четыре сложных типа данных:

- **DATE\_AND\_TIME (DT, Дата и время)** Дата и время (в формате BCD-числа);
- **STRING (Строка)** Строка литер длиной до 254 знаков;
- **ARRAY (Массив)** Переменная-массив (совокупность переменных одного типа);
- **STRUCT (Структура)** Переменная-структура (совокупность переменных разных типов).

Типы данных предопределяются пользователем при их использовании: задается длина в типе STRING (строка литер), сочетание и размер в типах ARRAY и STRUCT (структура).

## **3. Типы данных определенные пользователем (более чем 32 бита)**

**User data type - UDT** (Пользовательский тип данных) соответствует структуре (комбинация компонентов любых типов) с действием на глобальном уровне. Вы можете воспользоваться пользовательским типом данных, если в вашей программе часто фигурирует структурный тип и переменные, или вы хотите структуре данных присвоить имя.

Типы UDT обладают глобальным действием; то есть, они описываются один раз и доступны для использования во всех блоках.

При объявлении переменных определяются следующие их свойства:

- символическое имя
- тип данных
- видимость переменной

Переменные могут быть объявлены:

- в глобальной символьной таблице (элементарные типы данных)
- в таблице описаний глобального блока данных (все типы данных)
- в таблице описаний логического блока (OB, FB и FC)

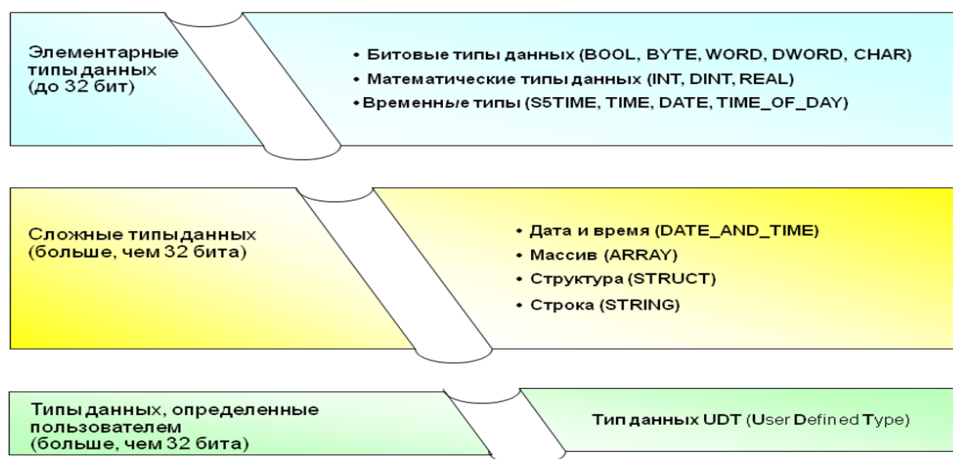


Рисунок 25 - Типы данных языка STEP 7

### Виды адресации

При написании программ в STEP 7 можно применять прямую адресацию или косвенная адресация (рис.26).

**Прямая адресация** может быть представлена в виде

- абсолютной адресации;
- символьной адресации.

**Абсолютная адресация** состоит из следующих основных полей – *идентификатор области памяти, идентификатор данных и адрес в этой области.*

Однако при большом числе переменных такая адресация неудобна, поэтому для придания смысловой нагрузки переменных вводятся их символьные обозначения, те применяется **символьная адресация.**

Для хранения символьных обозначений используется специальная таблица, содержащая четыре столбца, с названием, адресом, типом данных и комментарием. Символьное имя Symbol содержит до 24 символов, начинается с буквы, может содержать подчеркивания.

Например, если входной дискретный модуль занимает адреса от 0 до 3, то входы могут обозначаться как I 0.0, I 0.1 и т.д. Аналогично выходы для цифрового модуля вывода, который занимает адреса с 4 по 7, обозначаются как Q 4.0, Q 4.1 и т.д.

Косвенная адресация является более сложным видом адресации и в данном пособии не рассматривается.(данный метод адресации описан в [1]). Основные виды адресации доступные с STEP 7 представлены на рисунке .



Рисунок 26 - Виды адресаций доступные в языке STEP 7

### Обращение к данным в областях памяти

Контроллер S7 хранит информацию в различных местах памяти, которые имеют однозначные адреса. Программист можете явно указать адрес в памяти, к которому он хочет обратиться. Благодаря этому программа имеет прямой доступ к информации.

В контроллер S7 существуют следующие области памяти:

### Память входов образа процесса: обозначается - I

В начале каждого цикла S7-300 опрашивает физические входы и записывает полученные значения во регистр входов образа процесса. К образу процесса можно обратиться в формате бита, байта, слова и двойного слова:

Тип данных	Обращение	Пример
Бит	I[адрес байта]. [адрес бита]	I0.1
Байт	I[длина(B)][начальный адрес байта]	IB4
Слово	I[длина(W)][начальный адрес байта]	IW4
Двойное слово	I[длина(D)][начальный адрес байта]	ID4

### Память выходов образа процесса: Q

В конце цикла S7-200 копирует значения, хранящиеся в регистре выходов образа процесса, в физические выходы. К образу процесса можно обратиться в формате бита, байта, слова и двойного слова:

Тип данных	Обращение	Пример
Бит	Q[адрес байта]. [адрес бита]	Q0.1
Байт	Q[длина(B)][начальный адрес байта]	QB4
Слово	Q[длина(W)][начальный адрес байта]	QW4
Двойное слово	Q[длина(D)][начальный адрес байта]	QD4

### Область битовой памяти (меркерная память): M

Биты памяти (меркеры) можно использовать как управляющие реле для хранения промежуточных результатов операций или другой управляющей информации. К битам памяти можно обратиться в формате бита, байта, слова и двойного слова:

Тип данных	Обращение	Пример
Бит	M[адрес байта]. [адрес бита]	M10.2
Байт	M[длина(B)][начальный адрес байта]	MB100
Слово	M[длина(W)][начальный адрес байта]	MW200
Двойное слово	M[длина(D)][начальный адрес байта]	MD200

К данным в других областях памяти

- **Область памяти таймеров – T;**
- **Область памяти счетчиков – C;**
- **Аккумуляторам**

обращаются, указывая в качестве адреса идентификатор области и номер элемента.

Тип данных	Обращение	Пример
Область памяти таймеров	T[номер элемента]	T37
Область памяти счетчиков	C[номер элемента]	C5
Аккумуляторам	AC[номер элемента]	AC0



Рисунок 27 - Пример обращения к одному и тому же адресу в формате байта, слова и двойного слова

### Обращение к периферийным модулям

ПЛК S7-300, 400 позволяют непосредственно обратиться к входным и выходным периферийным модулям. Можно обратиться в формате бита, байта, слова и двойного слова:

Обращение к входным модулям:

Тип данных	Обращение	Пример
Бит	PI[адрес байта]. [адрес бита]	PI0.1
Байт	PI[длина(B)][начальный адрес байта]	PIB4
Слово	PI[длина(W)][начальный адрес байта]	PIW4
Двойное слово	PI[длина(D)][начальный адрес байта]	PID4

Обращение к выходным модулям:

Тип данных	Обращение	Пример
Бит	PQ[адрес байта]. [адрес бита]	PQ0.1
Байт	PQ[длина(B)][начальный адрес байта]	PQB4
Слово	PQ[длина(W)][начальный адрес байта]	PQW4
Двойное слово	PQ[длина(D)][начальный адрес байта]	PQD4

**Важно!** Следует учитывать, что непосредственно обращение к периферийным модулям требует гораздо больших затрат времени, так как связь с периферийными модулями осуществляется по последовательному интерфейсу.

## 5. Пользовательское программирование DP-интерфейса

Децентрализованная периферия, подключенная к системе SIMATIC S7, обрабатывается с точки зрения пользовательской программы так же, как и центральная периферия. Исключение составляет децентрализованная периферия, подключенная к CP342-5. Обмен данными с DP-Slave происходит через области отображения входного/выходного процесса (области I и Q) или с помощью прямого доступа к периферии из пользовательской программы (PI, PQ).

Для обработки и оценки процессных и диагностических сигналов в распоряжении имеются соответствующие интерфейсы и функции. Возможно также непосредственное параметрирование DP-Slave'a из пользовательской программы.

### 5.1. Пример программы управления

Рассмотрим, как в этом случае может быть организовано управление электроприводом по сети Profibus:




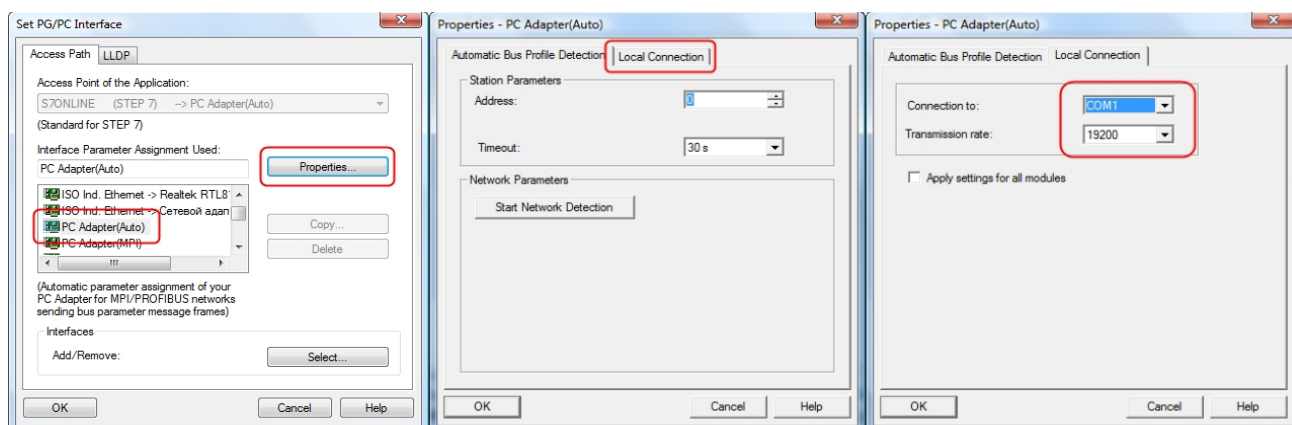
Для создания проекта нужно запустить приложение SIMATIC MANAGER. Все последующие действия будем рассматривать в по-шаговом режиме.

1. Для создания нового проекта нужно выполнить следующие действия:

- открыть мастер создания нового проекта, для этого нужно зайти в меню **File** и открыть '**New Project' Wizard...**;
- нужно выбрать тип центрального процессора, поддерживающего работу с децентрализованной периферией по сети Profibus, например **CPU315-2DP**. На этой же вкладке выбирается MPI адрес, который используется для установки связи между персональным компьютером и контроллером. Можно оставить значение по умолчанию, равное **2**;
- затем выбираются организационные блоки OB, которые будут в программе. Достаточно выбрать один организационный блок **OB1**, остальные блоки по мере необходимости можно добавить в процессе работы (см. таб. 16);
- выбираем язык программирования. Его также можно будет в дальнейшем изменить. Программа в примере будет написана на языке **LAD**;
- на заключительном этапе создания проекта предлагается задать имя проекта.

2. Настройка связи персонального компьютера с логическим контроллером производится в следующей последовательности (см. рис. 28):

- приложение для настройки соединения находится в меню **Options ® Set PC/PG interface...**;
- если контроллер подключен к компьютеру через COM порт, то нужно выбрать соединение PC Adapter (Auto) и через вкладку Properties... выбрать номер порта (рис. 26);
- если для связи используется порт USB, то в настройках соединения вместо COM порта нужно выбрать USB;
- для проверки связи с контроллером нужно нажать кнопку  - Download на панели управления для передачи созданной конфигурации в контроллер. В случае возникновения ошибки нужно проверить настройки соединения, в частности изменить номер COM порта.



**Рисунок 28 - Настройка соединения контроллера и персонального компьютера**

3. Конфигурирование Master-системы DP производится в следующей последовательности: Выберите DP master в слоте 2.1 и вставьте DP-master system рисунок 29. Теперь вы можете перемещать любые объекты и помещать их в master-систему, перетаскивая их при нажатой левой кнопке мыши.

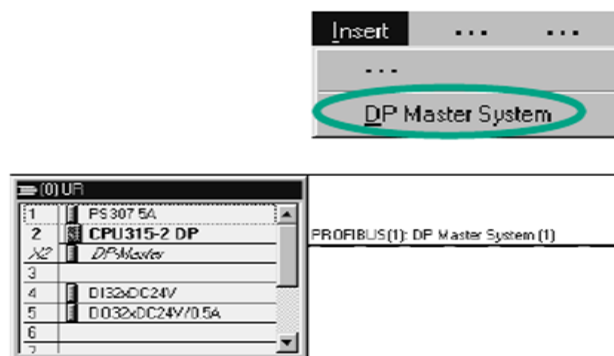


Рисунок 29 - Конфигурирование Master-системы DP

4. Добавление преобразователя частоты в сеть и его настройка осуществляются следующим образом:

- выбираем из списка оборудования преобразователь частоты Micromaster 440 и перетаскиваем его на созданную сеть PROFIBUS(1) (рис. 30);

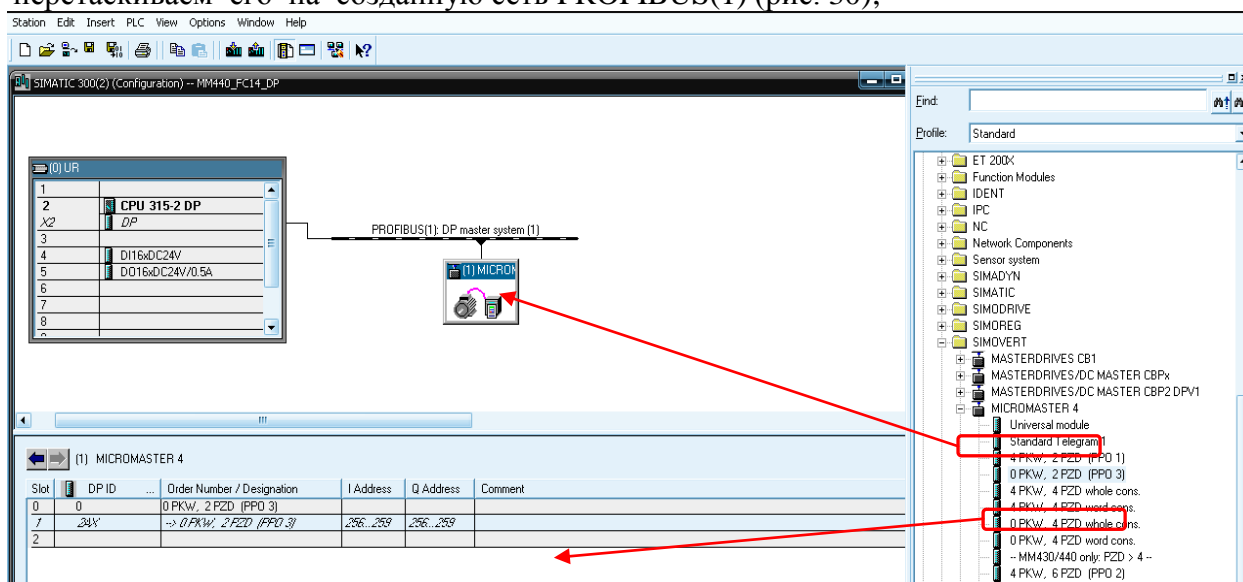


Рисунок 30 - Добавление преобразователя частоты Micromaster 440 в сеть PROFIBUS

- На следующем этапе выберем самый простой профиль PRO 3 для обмена данными между преобразователем и ПЛК (см. рис. 30). Этот самый простой вид телеграммы позволяет отправлять «слово управления 1» и «задание на скорость» по сети.

На этом этап настройки оборудования можно считать законченным и можно загрузить конфигурацию оборудования в логический контроллер с помощью кнопки **Download to Module**.

5. Произвести параметрирование преобразователя Micromaster 440 для управления по сети PROFIBUS. Для настройки лучше воспользоваться соответствующим руководством [1].

6. Управление пуском и реверсом преобразователя частоты с помощью «Слово управления 1» программируется следующим образом.

- Принцип управления преобразователем будет следующий: при одновременном включении либо отключение тумблеров SA1 (I124.0) и SA2 (I124.1) подается команда Стоп (47E hex). Когда тумблер SA1 (I124.0) будет включен, на преобразователь подается сигнал Пуск (47F hex). При этом блокируется одновременная отправка команд пуск вперед и пуск назад путем установки соответствующих блокировок. При подаче сигнала SA2 (I124.1), в «Слово управления 1» отправляется сигнал пуска в обратном направлении (C7F hex), при этом SA1 должен быть выключен. Все сигналы для управления преобразователем отправляются преобразователю с помощью команды MOVE (см. табл. 25).
- После этого сформируем сигнал задания на частоту вращения. Сделаем так, что при подаче сигнала SA1 (I124.0) или SA2 (I124.1) подается сигнал задания на частоту 40 Гц (3333 hex),

при подаче сигнала SA1 (I124.3) подается сигнал задания на частоту 25 Гц (2000 hex),  
 И этот сигнал командой MOVE отправляется в преобразователь. Программа приведена на рис. 31-33.

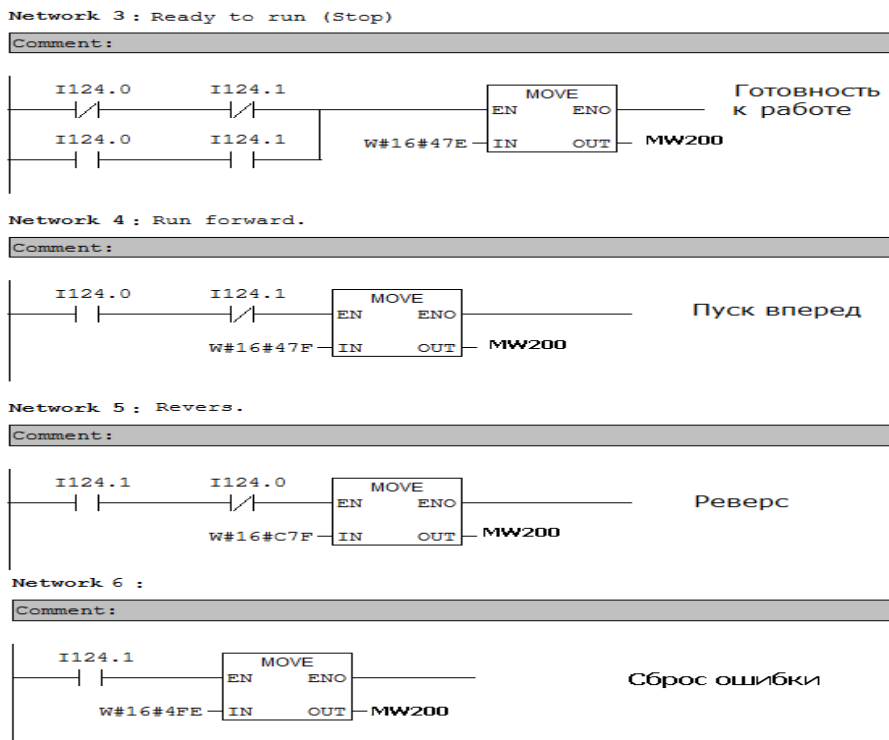


Рисунок 31 - Подача управляющих команд через «Слово управления 1»

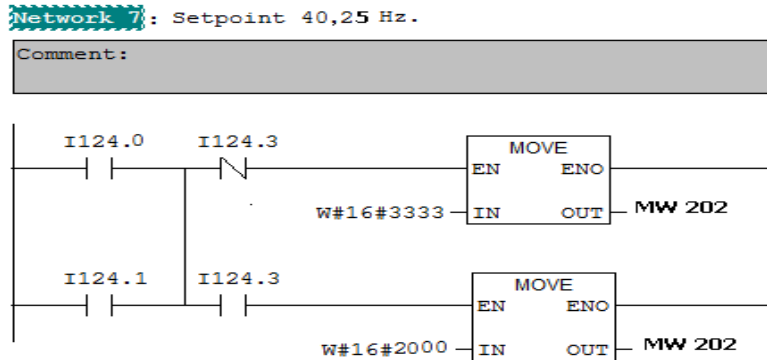


Рисунок 32 - Формирование сигнала задания на частоту



L MD 200  
 T PQD 256 //адрес ПЧ из HW

Рисунок 33 - Передача Слова управления и задания по скорости в ПЧ

## 6. Функции для обмена DP-пользовательскими данными и функции для сигналов от процесса

### 6.1. Обмен консистентными DP-данными с помощью SFC14 DPRD\_DAT и SFC15 DPWR\_DAT

Для обмена областями DP- данных, которые имеют консистентную (единую) структуру с размером 3 или более 4 байт, нельзя использовать обычные команды обращения к байту, слову или двойному слову. В этом случае обмен данными происходит с помощью системных блоков функций SFC14 DPRD\_DAT и SFC15 DPWR\_DAT.

#### SFC14 DPRD\_DAT

Область входных данных DP-Slave'a читается с помощью вызова **SFC14 DPRD\_DAT**. **SFC14** имеет приведенные в таблице 19 входные и выходные параметры, которые при вызове должны быть обеспечены соответствующим образом. Если DP-Slave имеет несколько консистентных входных модулей (блоков, областей), то должен для каждого из них использоваться отдельный вызов SEC14.

Таблица 19 - Параметры для SFC14 DPRD\_DAT

Параметр	Описание (тип доступа)	Тип данных	Область памяти	Описание
LADDR	INPUT	WORD	I,Q,M,D,L;const	Задание (в 16-ичном формате) спроектированного в HW-Config начального адреса входного модуля DP-Slave
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Возвращаемое значение SFC
RECORD	OUTPUT	ANY	I, Q, M, D, L	Область для прочитанных пользовательских данных

Внимание В этом параметре в типе данных ANY должен быть только тип BYTE.

#### Параметр RET\_VAL

В таблице 5.21 представлены коды ошибок, возвращаемые в параметре RET\_VAL функцией SFC14.

Таблица 20 - Значения параметра RET\_VAL у SFC14 DPRD\_DAT

Код ошибки W#16#...	Комментарий
000	Ошибок нет
8090	Для заданного логического базового адреса не спроектирован модуль или ограничение на длину консистентных данных не соблюдено (не учтено)
8092	В параметре RECORD тип данных отличен от BYTE
8093	Для заданного в LADDR логического адреса не существует DP-модуля, из которой Вы можете читать консистентные данные.
80A0	Данный модуль неисправен
80B0	Выход из строя Slave при внешнем DP-подключении.
80B1	Длина целевой области не равна длине полезных данных, спроектированных в HW-Config
808x 80B2 80B3 80C0 80C2 80Fх 87ху	Системная ошибка при внешнем DP-подключении

#### SFC15 DPWR\_DAT

Области выходных консистентных данных DP-Slave записываются с помощью вызова SFC15 DPWR\_DAT в DP-Slave. SFC15 имеет входные и выходные параметр, приведенные в таблице 21

Если DP-Slave имеет несколько консистентных выходных блоков, то перенос данных должен осуществляться для каждого такого блока вызовом SFC15.

**Таблица 21 - Параметры для SFC15 DPWR\_DAT**

Параметр	Описание (тип доступа)	Тип данных	Область памяти	Описание
LADDR	INPUT	WORD	I,Q,M,D,L;const	Задание (в 16-ичном формате) спроектированного в HW-Config начального адреса выходного модуля DP-Slave
RECORD	OUTPUT	ANY	I, Q, M, D, L	Область-источник для записываемых пользовательских данных
RET_VAL	OUTPUT	INT	I, Q, M, D, L	Возвращаемое значение SFC

### Описание параметров

#### Параметр RECORD

Параметр RECORD описывает область-источник для консистентных выходных данных, которые переносятся из S7-CPU в DP-Slave. Задание длины параметра RECORD должно соответствовать спроектированной в HW-Config длине выходного модуля DP-Slave.

Далее нужно обратить внимание, что этот параметр имеет тип ANY и допускает только тип данных BYTE.

#### Параметр RET\_VAL

В таблице 22 представлены коды ошибок SFC15, возвращаемые в параметре RET\_VAL.

**Таблица 22 - Коды специфических ошибок, возвращаемых в RET\_VAL для SFC15 DPWR\_DAT**

Код ошибки W#16#...	Комментарий
0000	Нет ошибки
8090	Для заданного логического базового адреса модуль не спроектирован или не соблюдено ограничение на длину консистентных данных
8092	В параметре RECORD (тип ANY) указан тип, не совпадающий с BYTE
8093	Для заданного в LADDR логического адреса не существует DP-модуля, в который Вы можете записывать консистентные данные
80A1	Выбранный модуль неисправен
80B0	Неисправный Slave во внешнем DP-подключении
80B1	Длина заданной области-источника не равна спроектированной в HW-Config длине пользовательских данных
80B2	Системная ошибка при внешнем подключении DP
80B3	Системная ошибка при внешнем подключении DP
80C1	Данные, предварительно записанные в модуль, еще не обработаны
808x 80Fx 85xy 80C2	Системная ошибка при внешнем DP- подключении

## 6.2. Пример программы управления

Рассмотрим, как в этом случаи может быть организовано управление электроприводом по сети Profibus:

Для создания проекта нужно запустить приложение SIMATIC MANAGER. Все последующие действия будем рассматривать в по-шаговом режиме.

Шаги 1-5 полностью соответствуют ранее рассмотренному примеру.

6. Для удобства работы создадим блок данных DB1:

· для этого нужно перейти в раздел Блоки в SIMATIC MANAGER и с помощью правой кнопки мыши создать блок данных **Data Blok** (рис. 34):

в блоке **DB1** будут храниться следующие данные:

- данные, которые отправляются в преобразователь (data), то есть «Слово управления 1 (CTW 1)» и «Задание на частоту (HSW)»;

- данные, которые отправляются в преобразователь область параметров (RXDATA)
- данные, которые будут передаваться от преобразователя область данных процесса (TXDATA).

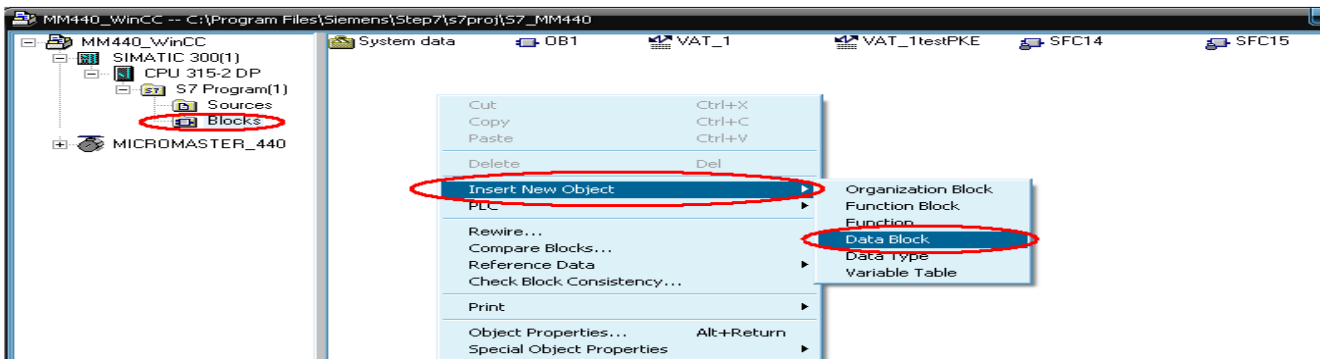


Рисунок 34 - Создание блока данных

Структура блока данных представлена на рисунке 35

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	DB_VAR	INT	0	Temporary placeholder variable
+2.0	data	STRUCT		
+0.0	STW1	INT	0	Управляющее слово 1
+2.0	HSW	INT	0	Главное задание
=4.0		END_STRUCT		
+6.0	RXDATA	STRUCT		
+0.0	PKE	INT	0	
+2.0	IND	INT	0	
+4.0	PWE1	INT	0	
+6.0	PWE2	INT	0	
=8.0		END_STRUCT		
+14.0	TXDATA	STRUCT		
+0.0	PKE	INT	0	
+2.0	IND	INT	0	
+4.0	PWE1	INT	0	
+6.0	PWE2	INT	0	
=8.0		END_STRUCT		
=22.0		END_STRUCT		

Рисунок 35 - Структура блока данных

7. После создания блока данных можно переходить к написанию основной программы для управления преобразователем в блоке OB1:

· создадим две стандартные системные функции SFC14 и SFC15 для обмена данными между логическим контроллером и приводом. Для этого нужно открыть стандартную библиотеку STEP 7 через меню File → Open и скопировать стандартные функции SFC14 и SFC15 в проект (см. рис. 36);

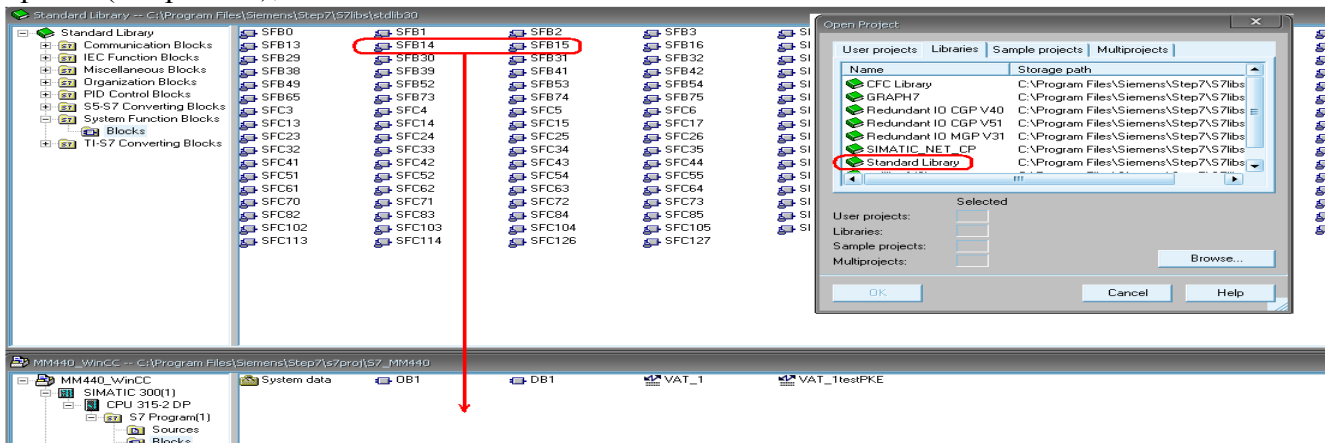


Рисунок 36 - Копирование системных функций SFC14 и SFC15 из стандартной библиотеки

8. Создадим программу управления в блоке OB1, поскольку этот блок имеет собственную статическую память, часть ее будем использовать для хранения переменных. Для этого, в разделе описания статической памяти блока, объявим следующие переменные:

- TXDATA типа -Structur;
- PKE типа – Int;
- IND типа – Int;
- PWE1 типа – Int;
- PWE2 типа – Int.

Переменные будут использоваться при формировании слов управления и слов состояния (рисунок 37)

TEMP0	Bool	20.0	
TXDATA	Struct	22.0	
PKE	Int	22.0	
IND	Int	24.0	
PWE1	Int	26.0	
PWE2	Int	28.0	

**Рисунок 37 - Временные переменные блока OB1**

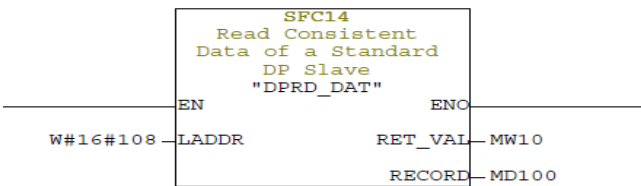
Программа блока OB1 будет иметь следующий вид рисунок 38:

Network: 1

---

Прием 2х слов от MM440 (PZD1 и PZD2): Слова состояния1 MW100 и выходная частота MW102

---

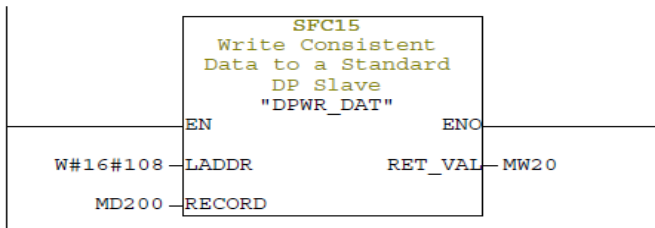


Network: 2

---

Передача в MM440 двух слов (PZD1 и PZD2): Слова управления1 MW200 и Задание на частоту MW202

---



Network: 3

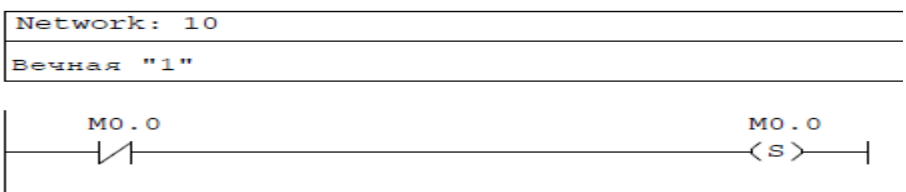
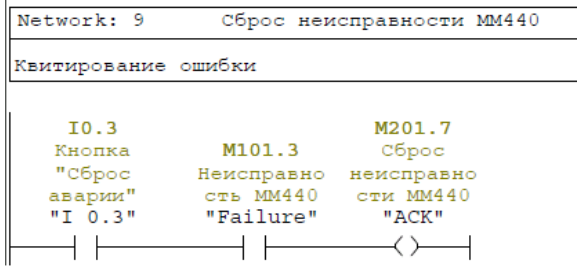
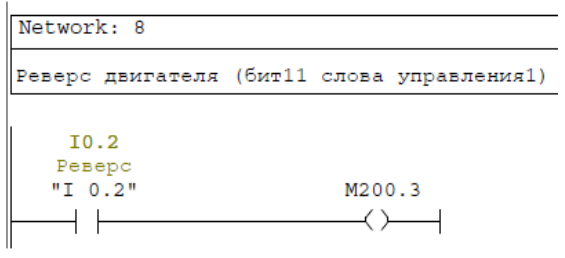
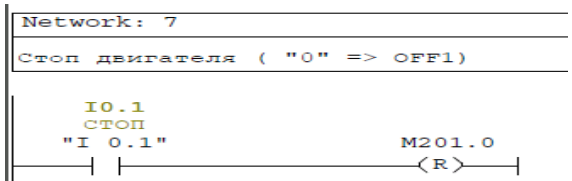
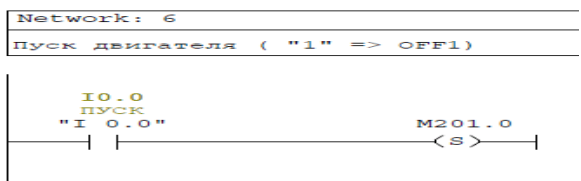
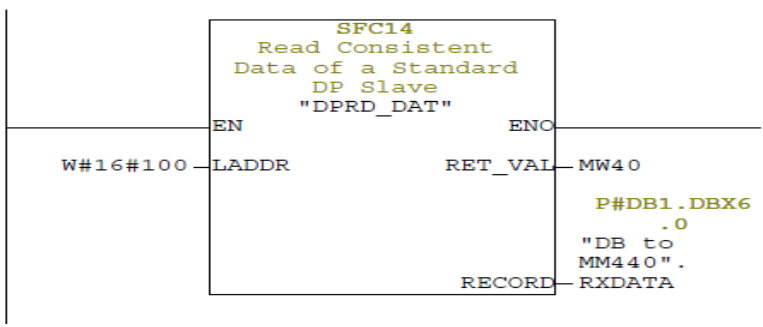
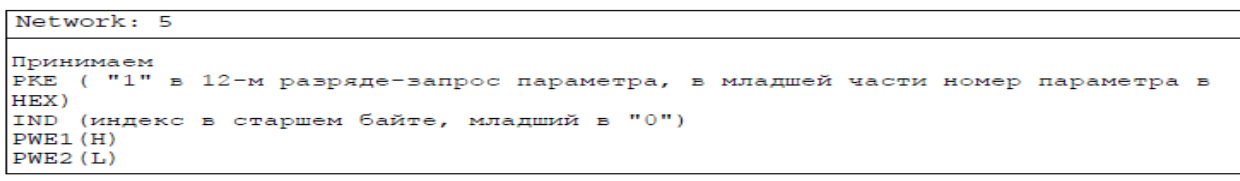
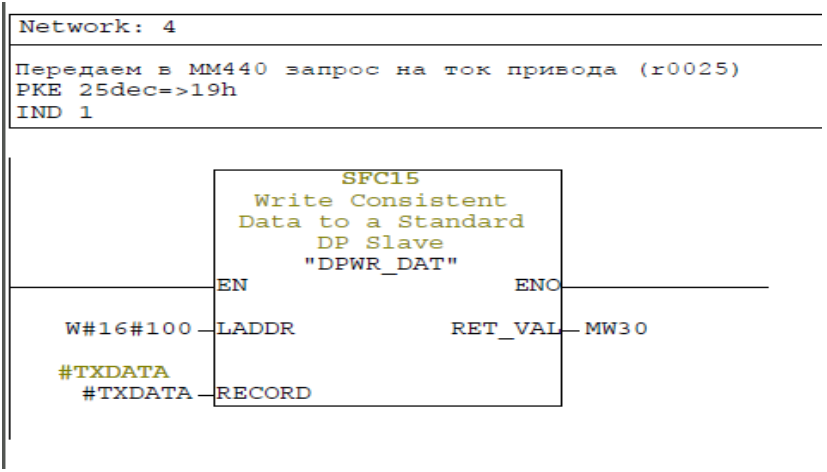
---

PKE

---

L	"DB to MM440".TXDATA.PKE	DB1.DBW14
T	#TXDATA.PKE	#TXDATA.PKE
L	"DB to MM440".TXDATA.IND	DB1.DBW16
T	#TXDATA.IND	#TXDATA.IND
L	"DB to MM440".TXDATA.PWE1	DB1.DBW18
T	#TXDATA.PWE1	#TXDATA.PWE1
L	"DB to MM440".TXDATA.PWE2	DB1.DBW20
T	#TXDATA.PWE2	#TXDATA.PWE2







```

Network: 12
Задание на скорость

L "заданная скорость" //установка скорости [Hz] MD50 -- заданная скорость
L 0.000000e+000
<R // установка скорости >0Hz
JC N004
L "заданная скорость" MD50-- заданная скорость
L 1.000000e+002
>R // установка скорости < 100Hz ?
JC N004
L "заданная скорость" MD50-- заданная скорость
L 1.000000e+002 // для сохранения 2 знаков после запятой
*R
RND //conversion of a 32-bit IEEE floating-point number to 32-bit
integer
L 16384 // 16384 = 4000HEX = 50Hz
*D
L 5000
/D
T "Задание на скорость" MW202
N004: NOP 0

```

```

Network: 13 Масштабирование скорости для вывода на индикатор
100Гц - 8000hex или 32768 (дес)

L "Скорость дв-ля" MW102
ITD
DTR
L 1.000000e+002
*R
L 3.276800e+004
/R
RND
T MW 300

```

Рисунок 38 - Блок ОВ 1 программы

### 6.3. Пример программы управления с использованием свободной конфигурация циклических данных

Последовательность действий, для организации управления преобразователем частоты по сети Profibus, полностью соответствует последовательности предложенной в

Для создания проекта нужно запустить приложение SIMATIC MANAGER. Все последующие действия будем рассматривать в по-шаговом режиме.

Шаги 1-5 полностью соответствуют ранее рассмотренному примеру.

#### 6.3.1. Определение адресации устройств на шине Profibus

Важным условием ввода в эксплуатацию коммуникационного модуля PROFIBUS является задание адреса сети PROFIBUS. Установка адреса выполняется одним из двух способов (см.п2.1):

- с помощью наборных переключателей DIP, расположенных на коммуникационном модуле;
- через параметр «P0918».

#### **ВАЖНО!**

Функция «сброс параметров преобразователя на заводскую настройку» сбрасывает также и адрес PROFIBUS на значение 3, если он был установлен через параметр «P0918».

#### 6.3.2. Конфигурирования параметров связи для преобразователя

Прежде всего, необходимо осуществить выбор вида управления преобразователем.

Для этого необходимо установить параметр «P0719» = 66. Управление преобразователем по PROFIBUS. При этом, коммуникационный модуль, получив первое слово управления ZSW1, копирует данные в регистр «P2090» оперативной памяти преобразователя частоты. Второе и третье слово управления будут скопированы в регистр «P2050» по адресам «P2050:1» и «P2050:2». Четвертое, дополнительное слово управление ZSW2, будет скопировано в оперативную память преобразователя в регистр «P2091». Пятое, шестое, седьмое и восьмое слово соответственно будут доступны по адресам «P2050:4», «P2050:5», «P2050:6», «P2050:7».

В параметре «P2040» определяется, будет ли преобразователь контролировать доставку пакетов телеграмм PROFIBUS по условиям заданного таймаута.

- «P2040» = 0 означает отсутствие контроля
- «P2040» > 0 означает, что значение параметра «P2040» и есть время отсутствия телеграммы в миллисекундах. (По умолчанию параметр имеет значение =20 мс)

Если, параметр «P2040» > 0, и за контрольное время отсутствия телеграммы модуль связи PROFIBUS не принял никаких новых телеграмм, появляется сообщение о сбое «F0070» и преобразователь частоты останавливает привод.

Если, параметр «P2040» = 0 никаких действий не выполняется.

#### **ВАЖНО!**

При отключении контроля параметров связи, преобразователь становится неуправляемым.

#### 6.3.3. Определения необходимых параметров требуемых для контроля ПЧ при проектировании исходящей информации от преобразователя.

Исходящая информация формируется путем задания номера регистров преобразователя частоты в индексы регистра «r2051», предназначенного для формирования исходящей телеграммы.

В преобразователе Micromaster 440 нет жесткой последовательности задания параметров в исходящую телеграмму. Тем не менее, по стандарту IEC 61800-7, и соглашению между пользователем и производителем первым словом телеграммы рекомендуется задавать первое слово состояния STW1. Для чего в параметре «r2051», индекс [0], присваивается значение

«52» означающее использование регистра «r0052» - первое слово состояния.

Пример формирования исходящей телеграммы в 8 слов с использованием наиболее распространенных параметров для передачи в контроллер представлен на рисунке39:

- r2051[0]=:52 = r0052 слово состояния STW1;
- r2051[1]=:21 = r0021 текущая частота на выходе инвертора;
- r2051[2]=:722 = r0722 текущее положение дискретных входов инвертора;
- r2051[3]=:26 = r0026 значение напряжения звена постоянного тока;
- r2051[4]=:27 = r0027 текущий ток на выходе инвертора;
- r2051[5]=:32 = r0032 текущая мощность на выходе инвертора;
- r2051[6]=:39 = r0039 счетчик потребления электроэнергии (кВт\*ч);
- r2051[7]=:31 = r0031 момент на выходе инвертора;

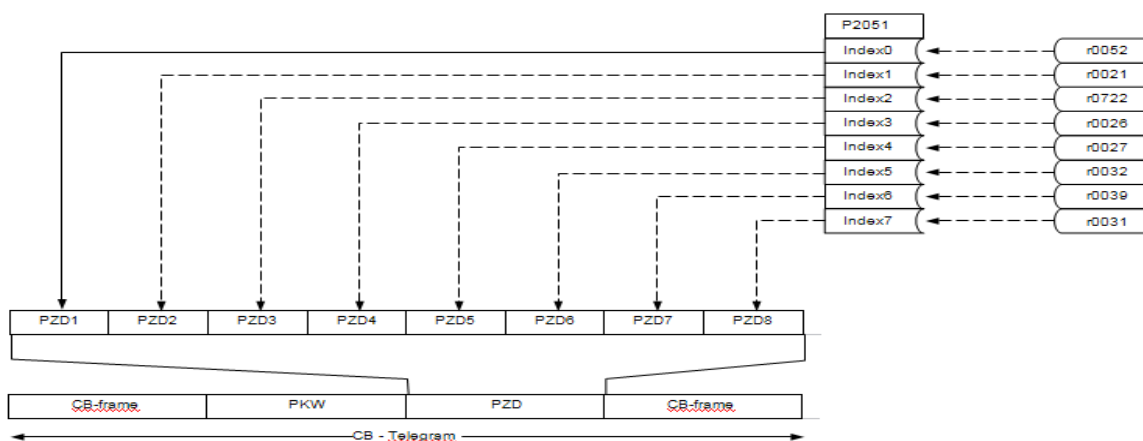


Рисунок 39 - Пример 1. Формирование исходящей телеграммы в 8 слов

Соответственно, для исходящей телеграммы в 4 слова, возможно, использовать индексы от [0] до [3](см. рис 40):

- r2051[0]=:52 = r0052 слово состояния STW1;
- r2051[1]=:21 = r0021 текущая частота на выходе инвертора;
- r2051[2]=:26 = r0026 значение напряжения звена постоянного тока;
- r2051[3]=:27 = r0027 текущий ток на выходе инвертора;

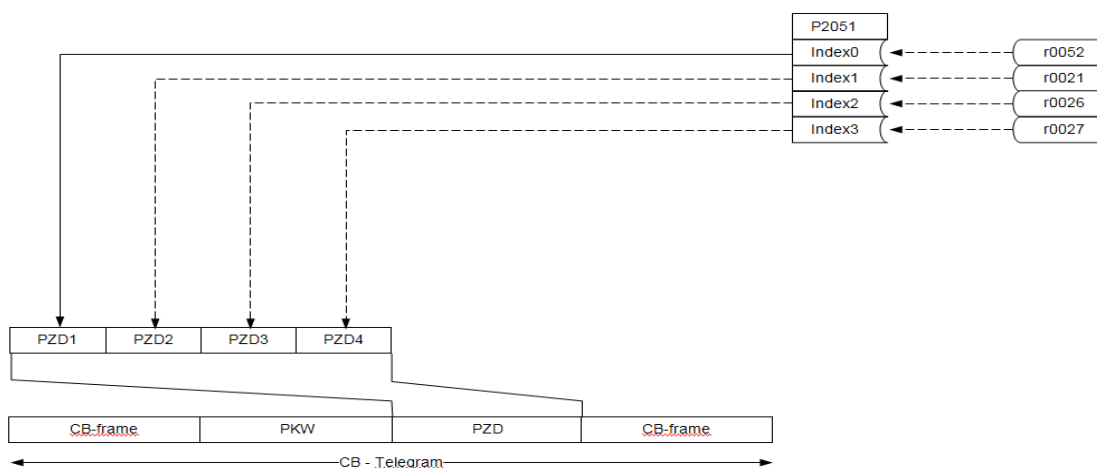


Рисунок 40 - Пример 2. Формирование исходящей телеграммы в 4 слова

Рекомендуется использовать чтение и передачу в контроллер параметра «r0026» (значение напряжения звена постоянного тока), для контроля наличия электропитания на входе преобразователя.

### 6.3.4. Проектирование использования входящей информации в преобразователе

Входящая информация поступает в регистр «r2050», в соответствующие индексы до 8 слов. Для преобразователей Micromaster 440 данные регистра «r2050» в индексах [0] и [3] копируются в область оперативной памяти преобразователя, по адресам «r2090» и «r2091». К регистрам «r2090» и «r2091» можно обращаться побитно, для использования соответствующих битов в качестве команд управления. Поэтому, при формировании телеграммы в контроллере, слово управления ZSW1 необходимо задавать первым, а слово управление ZSW2 четвертым. Остальные слова, формируются в контроллере в зависимости от предпочтений пользователя. Считывание информации производится из регистра «r2050» из соответствующего индекса, например:

Уставка ПИД-регулятору «P2253»=: r2050[1]

Принцип использования битов команд слова управления ZSW1 приведен на рисунке 41.

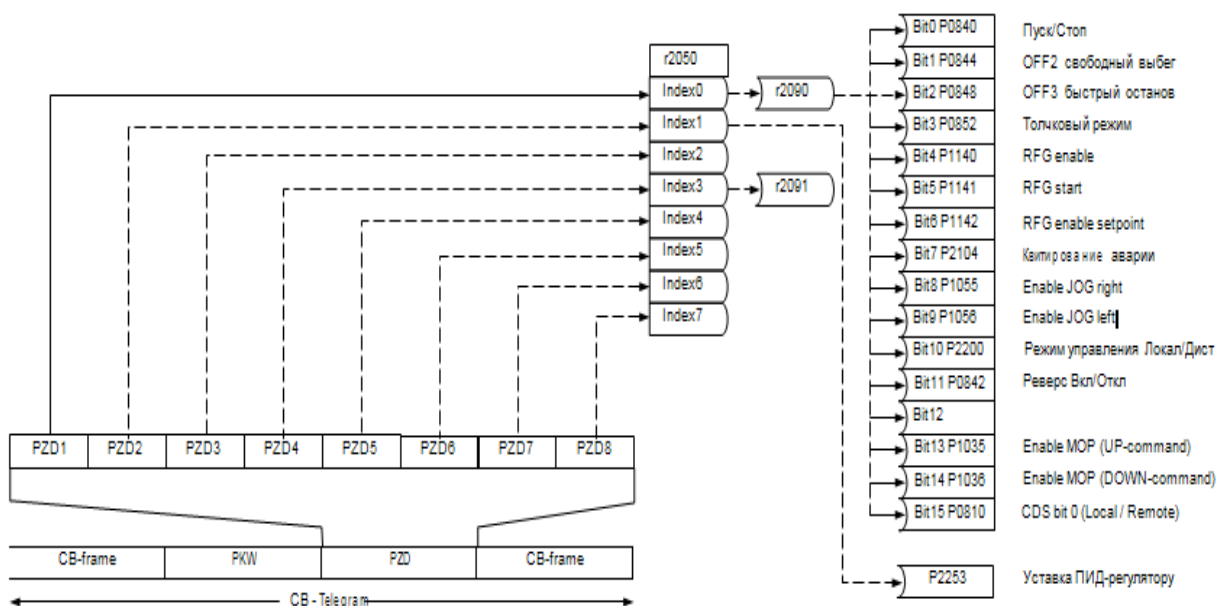


Рисунок 41 - Использование бит-команд поступающих в регистр P2050 из PLC

Из рисунка видно, что не требующиеся к использованию как биты слова управления ZSW1, так и слова в индексах от [2] до [7] регистра «r2050» не задействуются.

Бит 10 слова управления ZSW1 «Режим управления Локальный/Дистанционный» используется программой преобразователя без дополнительного указателя и при управлении по PROFIBUS должен быть установлен в 1 - «режим управления: Дистанционный».

При разработке структуры данных обмена между контроллером и преобразователем, необходимо точно представлять, какие данные нужны для нормальной работы устройства и как должны выполняться воздействия на слово управления ZSW1. (см. Таблицу 4)

**Поступающие в преобразователь команды, в двоичном виде, можно просмотреть в параметре «P2090».**

### 6.3.5. Конфигурирование и программирование ПЛК с использованием Step7.

Конфигурирование преобразователя в программе HWConfig было уже описано в предыдущих примерах, поэтому далее рассматривается коротко.

Вставка в проект преобразователя частоты выполняется путем перетаскивания графической иконки ПЧ на шину PROFIBUS. При этом в выпадающем окне необходимо задать адрес ПЧ в сети, и указать к какой сети PROFIBUS будет подключен. Затем необходимо перетащить нужный профиль в Rack преобразователя (рис.42)

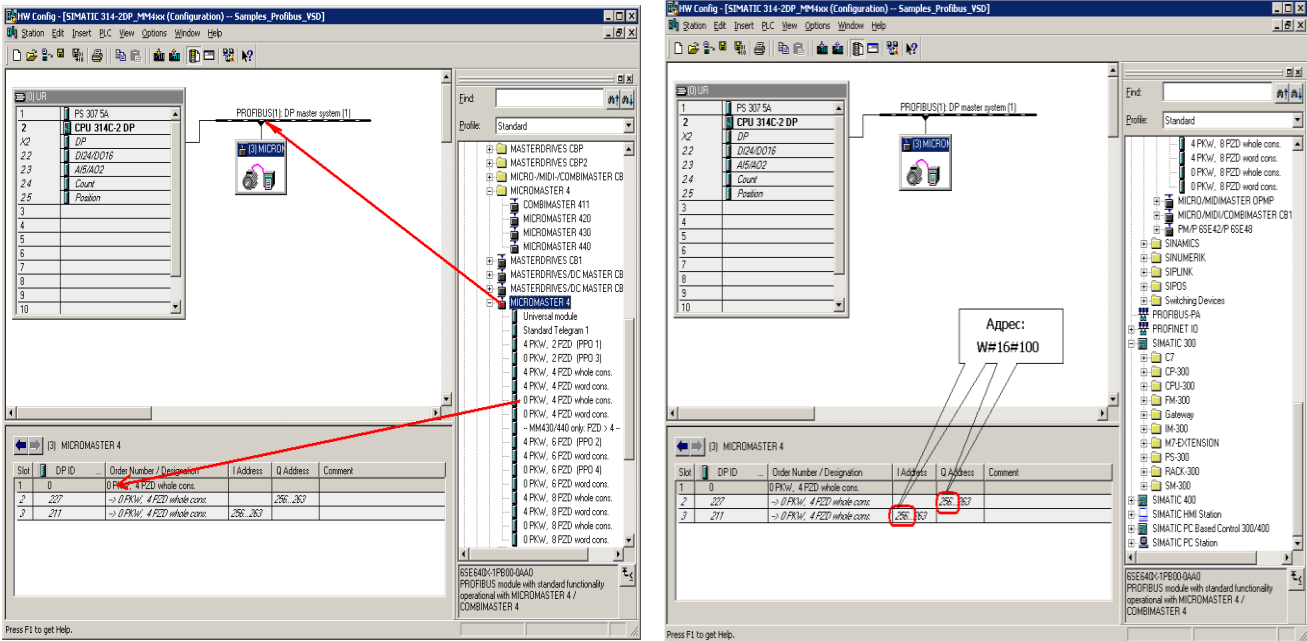


Рисунок 42 - Последовательность конфигурирование преобразователя в программе HWConfig

### 6.3.6. Разработка функции обмена данными с ПЧ в Step7

Создаем блок данных DB301

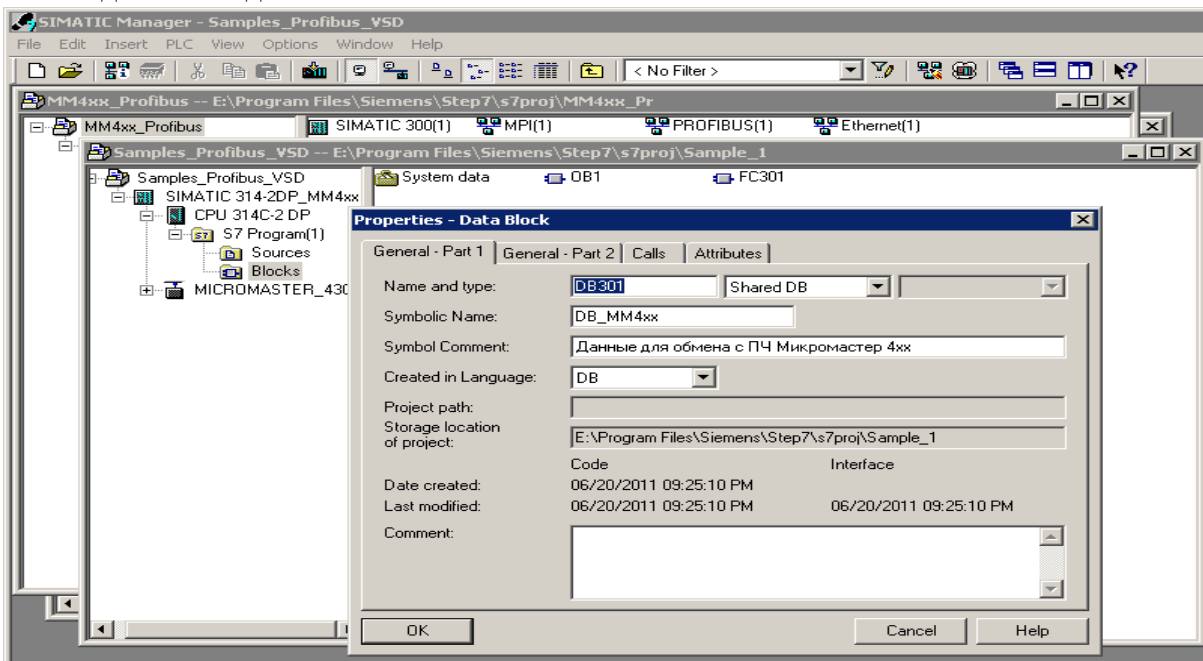


Рисунок 43- Порядок создания блока данных



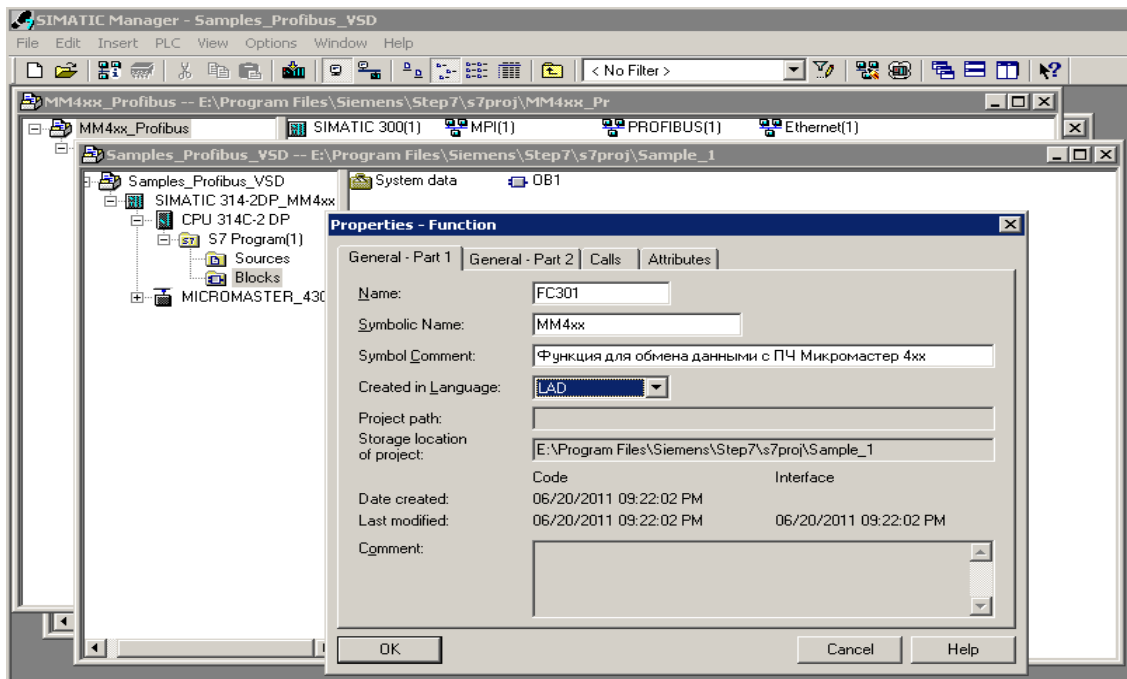
В блоке описываем структуру входящей и исходящей телеграммы. **Обратите внимание, что байты в словах STW1 и ZSW1 переставлены местами** (рис. 43).

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	DEVIATION_SETPOINT	BOOL	FALSE	Бит 08 Рассогласование уставки и фактического значения 0 ДА Deviation setpoint
+0.1	ASU_WORK	BOOL	FALSE	Бит 09 Управление AG (PZD-управление) 1 ДА PZD control 0 NO 1 YES
+0.2	MAX_FREQUENCY	BOOL	FALSE	Бит 10 Достигнута максимальная частота 1 ДА Maximum frequency reached 0 NO 1 YE
+0.3	MAX_CURRRNT	BOOL	FALSE	Бит 11 Предупреждение: граничное значение тока двигателя 0 ДА Warning: Motor cu
+0.4	TORNOZ_ACTUAL	BOOL	FALSE	Бит 12 Тормоз двигателя активен 1 ДА Motor holding brake active 0 NO 1 YES
+0.5	OVERLOAD_PRIVOD	BOOL	FALSE	Бит 13 Перегрузка двигателя 0 ДА Motor overload 0 YES 1 NO
+0.6	PRIVOD_RIGHT	BOOL	FALSE	Бит 14 Правое вращение 1 ДА Motor runs right 0 NO 1 YES
+0.7	OVERLOAD_INVERTER	BOOL	FALSE	Бит 15 Перегрузка вентиляционного преобразователя частоты 0 ДА Inverter overload 0
+1.0	GOTOV_K_VKL	BOOL	FALSE	Бит 00 Готов к включению 1 ДА Drive ready 0 NO 1 YES
+1.1	GOTOV_K_RABOTE	BOOL	FALSE	Бит 01 Готов к работе 1 ДА Drive ready to run 0 NO 1 YES
+1.2	PRIVOD_V_RABOTE	BOOL	FALSE	Бит 02 Работа/деблокирование импульсов 1 ДА Drive running 0 NO 1 YES
+1.3	ALARM_ACTUAL	BOOL	FALSE	Бит 03 Неисправность активна 1 ДА Drive fault active 0 NO 1 YES
+1.4	STOP2_PRIVOD	BOOL	FALSE	Бит 04 ВМКЛ2 активно 0 ДА OFF2 active 0 YES 1 NO
+1.5	STOP3_PRIVOD	BOOL	FALSE	Бит 05 ВМКЛ3 активно 0 ДА OFF3 active 0 YES 1 NO
+1.6	BLOK_ACTUAL	BOOL	FALSE	Бит 06 Блокировка включения активна 1 ДА ON inhibit active 0 NO 1 YES
+1.7	WARNING_ACTUAL	BOOL	FALSE	Бит 07 Предупреждение активно 1 ДА Drive warning active 0 NO 1 YES
+2.0	FREQUENCY	WORD	W#16#0	Текущая частота двигателя
+4.0	CURRENT	WORD	W#16#0	Текущий ток двигателя
+6.0	VOLTAGE_DC	WORD	W#16#0	DC напряжение (если его нет - значит нет входной напруги)
+8.0	CLOCKW_INCHING	BOOL	FALSE	Бит 08 JOG вправо для Micromaster и Sinamics не используется 0 НЕТ 1 ДА
+8.1	CCLOCKW_INCHING	BOOL	FALSE	Бит 09 JOG влево для Micromaster и Sinamics не используется 0 НЕТ 1 ДА
+8.2	ASU_UPRAVLENIE	BOOL	TRUE	Бит 10 Управление от контроллера (AG) 0 НЕТ 1 ДА
+8.3	REVERCE	BOOL	FALSE	Бит 11 Реверсирование 0 НЕТ 1 ДА
+8.4	RESERVED	BOOL	FALSE	Бит 12 RESERVED
+8.5	UP_FREQ	BOOL	FALSE	Бит 13 Потенциометр двигателя вверх 0 НЕТ 1 ДА
+8.6	DOWN_FREQ	BOOL	FALSE	Бит 14 Потенциометр двигателя вниз 0 НЕТ 1 ДА
+8.7	LOCAL_REMOTE_CDS	BOOL	FALSE	Бит 15 CDS Бит (Local/Remote) 0 НЕТ 1 ДА
+9.0	PUSK_PRIVOD	BOOL	FALSE	Бит 00 ВКЛ / ОТКЛ 0 НЕТ 1 ДА
+9.1	BREAK2	BOOL	TRUE	Бит 01 ОТКЛ2 : Электр. останов. 0 ДА 1 НЕТ
+9.2	BREAK3	BOOL	TRUE	Бит 02 ОТКЛ3 : Быстрая остановка 0 ДА 1 НЕТ
+9.3	DEBLOCK_IMPULSE	BOOL	TRUE	Бит 03 Команда на включение/отключение операционного режима 0 НЕТ 1 ДА
+9.4	DEBLOCK_RAZGON	BOOL	TRUE	Бит 04 Разрешение задатчика интенсивности (уставки) 0 НЕТ 1 ДА
+9.5	ENABLE_RFG	BOOL	TRUE	Бит 05 Фиксация задатчика интенсивности (уставки) 0 НЕТ 1 ДА
+9.6	DEBLOCK_SETPOINT	BOOL	TRUE	Бит 06 Деблокировка заданной величины 0 НЕТ 1 ДА
+9.7	KVIT_ALARM	BOOL	FALSE	Бит 07 Квитирование сбоя 0 НЕТ 1 ДА
+10.0	SETPOINT	WORD	W#16#0	Уставка
+12.0	RESEREVED_3	WORD	W#16#0	Резерв 1
+14.0	RESEREVED_4	WORD	W#16#0	Резерв 2
+16.0		END_STRUCT		

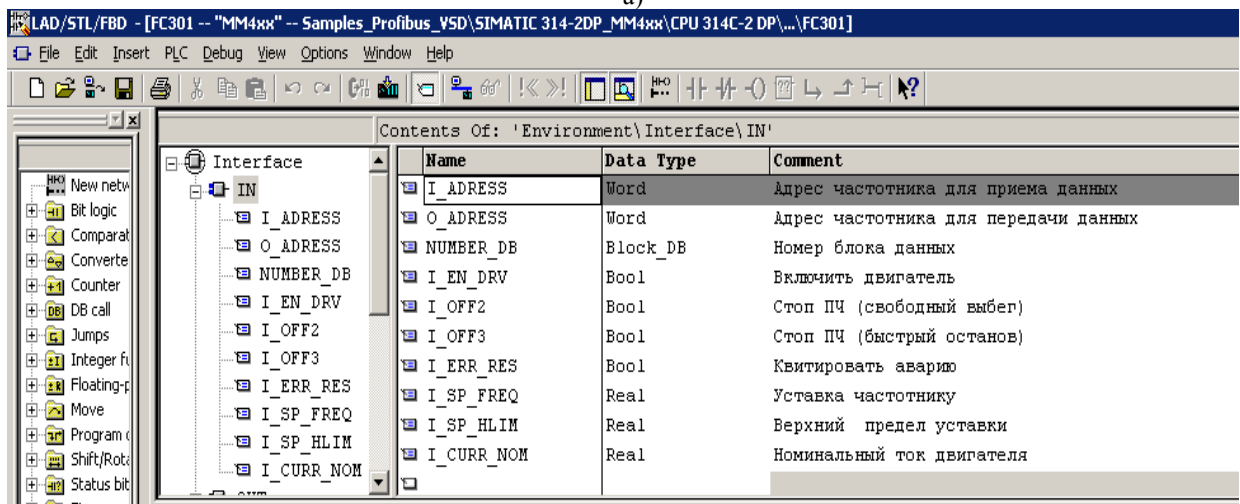
Рисунок 44 - Описание структуры блока DB301 (входящей и исходящей телеграмм)

Создаем функцию FC301 и формируем входные, выходные и временные переменные для данной функции рисунки (44 а,б,в,г).

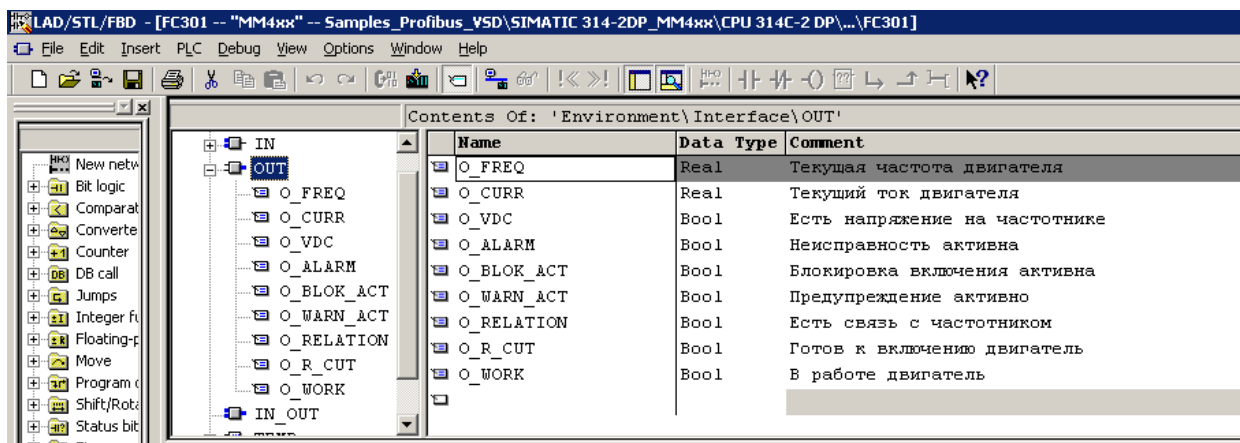




a)



б)



в)

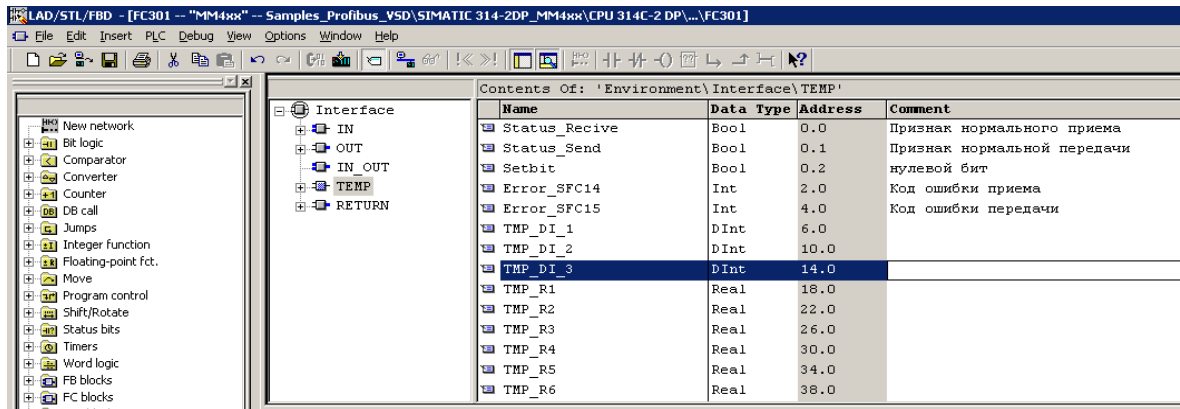


Рисунок 45 - Создание функцию FC301 (а), формирование входных (б), выходных (в) и временные переменных (г) для функции FC301.

Разрабатываем функцию обмена данными с ПЧ. Выполняем открытие блока данных DB301:

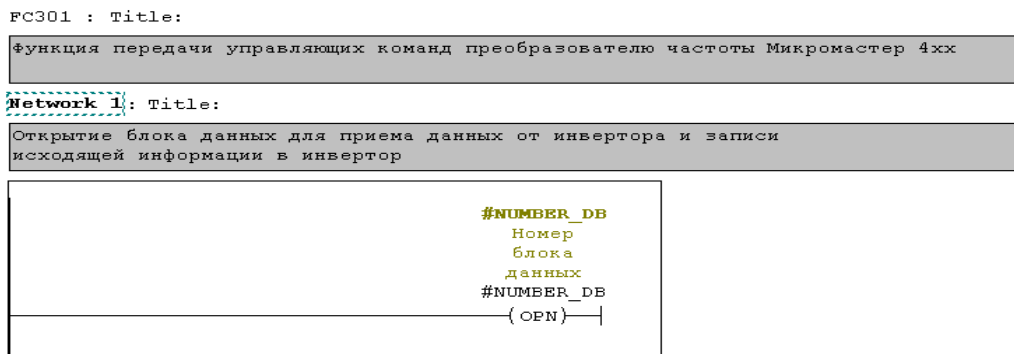


Рисунок 46 - Открытие блока данных DB301

Вставляем SFC14 (рис. ) и присваиваем:

- Начальный адрес области ввода на вход функции LADDR
- Возвращаемое значение функции RET\_VAL если 0, то работа без ошибок, если 1 то ошибка чтения данных
- Указатель для записи полученных данных в DB301

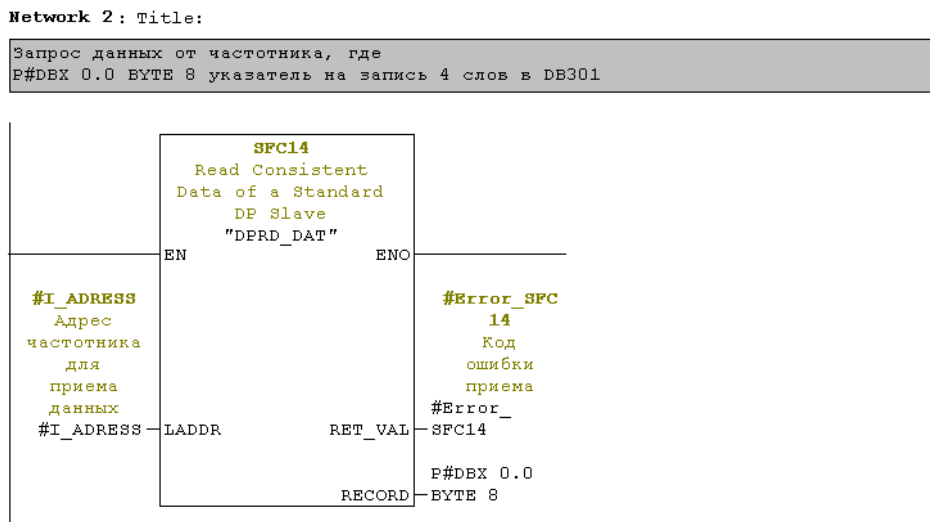


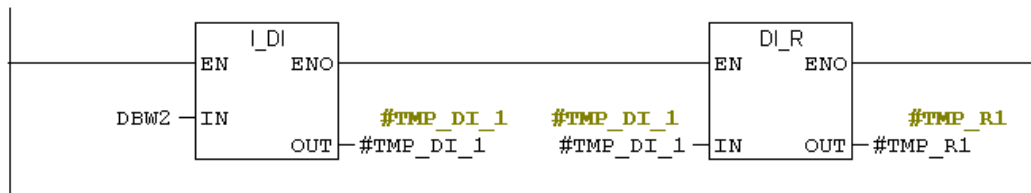
Рисунок 47 - Назначение параметров для блока приема данных по сети Profibus SFC14

Принятое и размещенное по адресу DB301.DBW2 слово содержит значение текущей выходной частоты. Производим конвертацию слова в значение REAL и масштабирование данной переменной по формуле (рис. ):

$$X1=(\text{знач}/16384)*50$$

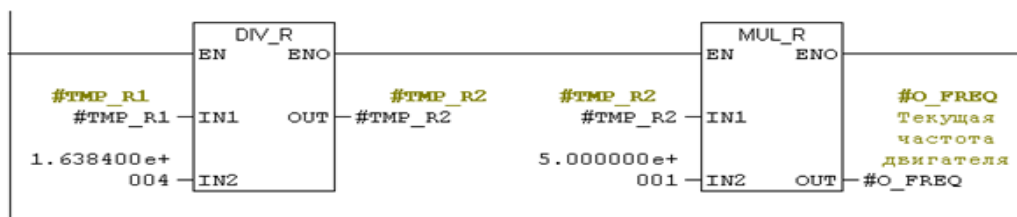
**Network 3:** Title:

Масштабирование текущей частоты двигателя  
 Конвертируем слово №2 текущей частоты в двойное целое, затем в плавающую точку



**Network 4:** Title:

Полученную величину делим на диапазон HEX=4000 (DEC=16384) и умножаем на 50 Гц



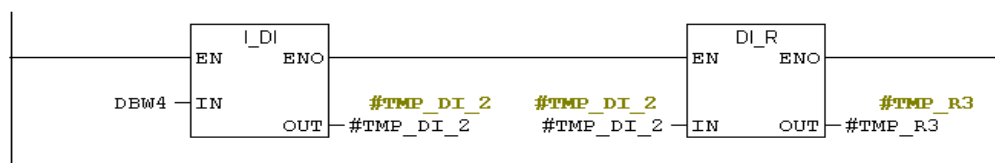
**Рисунок 48 –Масштабирование принятых данных о скорости**

Аналогично поступаем с принятым значением тока. Производим конвертацию текущего тока нагрузки из слова в REAL Масштабирование данной переменной по формуле (рис. ):

$$X1=(\text{знач}/8192)*I_{ном}$$

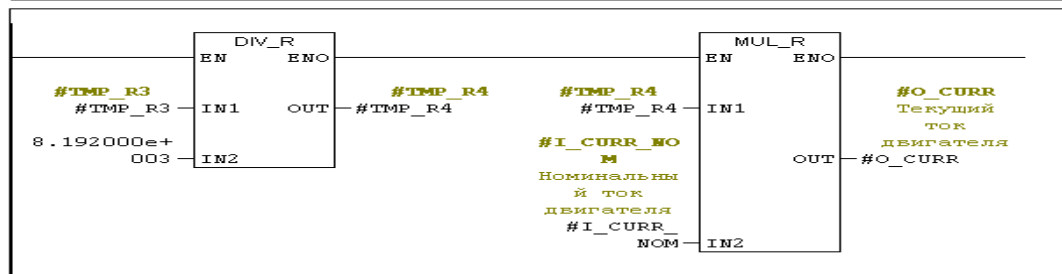
**Network 5:** Title:

Масштабирование текущего тока нагрузки двигателя  
 Конвертируем слово №3 текущей частоты в двойное целое, затем в плавающую точку



**Network 6:** Title:

Полученную величину делим на диапазон HEX=2000 (DEC=8192) и умножаем на значение номинального тока двигателя



**Рисунок 49 –Масштабирование принятых данных о токе нагрузки**

Сравниваем значение напряжения VDC во внутреннем контуре ПЧ с нормальным порогом, если напряжение ниже, значит, отсутствует питание преобразователя

Network 7 : Title:

Обработка значения напряжения VDC во внутреннем контуре частотника  
Слово №4

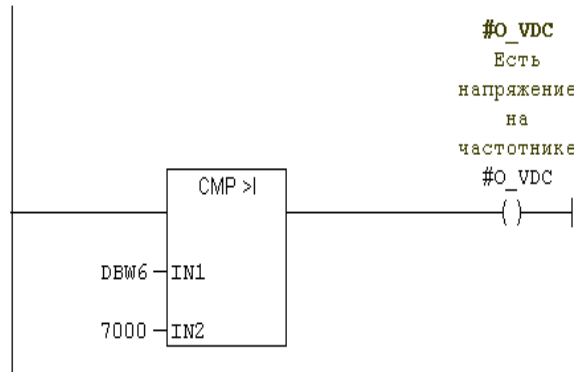


Рисунок 50 - Обработка значения постоянного напряжения на звене постоянного тока  
Обрабатываем значение RET\_VAL SFC14, если =0 то статус состояния связи =:1

Network 8 : Title:

Обработка признака ошибки связи при приеме данных

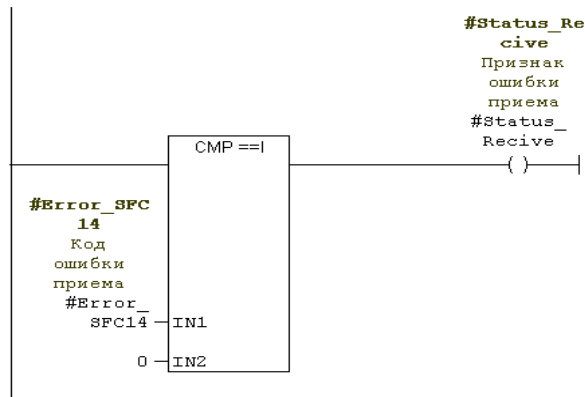


Рисунок 51 - Проверка отсутствия ошибок связи

Обрабатываем полученное слово состояния STW1 и проводим анализ битов готовности ПЧ к работе:

Network 9 : Title:

Обработка слова состояния (№1)  
Анализируются биты:  
"Готов к включению"  
"Готов к работе"

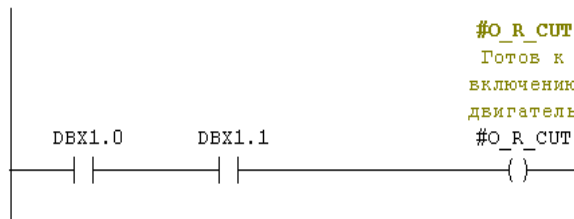


Рисунок 52 - Анализ битов готовности ПЧ к работе:

Считываем необходимые биты слова состояния и формируем выходные переменные функции:

**Network 10 : Title:**

Анализ бита "Двигатель в работе"



**Network 11 : Title:**

Анализ бита "Неисправность активна"



**Network 12 : Title:**

Анализ бита "блокировка включения активна"



**Network 13 : Title:**

Анализ бита "предупреждение активно"



**Рисунок 53 – Формирование выходных переменных функции**

Формируем слово управления ZSW1 и устанавливаем требуемые биты слова управления по умолчанию в 1:

**Network 14 : Формирование слова управления**

Обработка команды на включение



**Network 15 : Title:**

Останов двигателя выбегом  
Обратите внимание - инвертированная команда



**Network 16 : Title:**

Быстрый останов  
Обратите внимание - инвертированная команда



**Network 17 : Title:**

Квитирование аварии



Установка битов слова управления по умолчанию в 1

SET		//Установить в 1 следующие биты слова управления:
=	DBX 9.3	//Команда на включение операционного режима
=	DBX 9.4	//Разрешения значения интенсивности
=	DBX 9.5	//Фиксация значения задатчика интенсивности
=	DBX 9.6	//Деблокирование заданной величины
=	DBX 8.2	//Включить управление от контроллера

**Рисунок 54 Формирование битов слова управления ZSW1**

Выполняем конвертацию уставки частоты для передачи в ПЧ. Рекомендуется устанавливать верхний порог уставки в процентах. Т.е. 50Гц=100% .

Network 19: Title:

Масштабирование данных для уставки на частотник

```
L   #I_SP_FREQ           //загрузить уставку в формате плавающей точки
L   #I_SP_HLIM           //загрузить верхний предел значения датчика
/R                               //поделить значения
L   1.638400e+004        //загрузить диапазон HEX=4000 (DEC=16384)
*R                               //перемножить
TRUNC                          //выделить целое число
T   DBW 10                //загрузить в слово DBW10 для передачи в частотник
```

Рисунок 55 - Масштабирование данных заданной скорости

Организуем передачу слова управления в ПЧ. Используем для этого SFC15 и присваиваем:

- Начальный адрес области вывода на вход функции LADDR
- Вводим указатель для чтения данных из DB301
- Возвращаемое значение функции RET\_VAL если 0, то работа без ошибок, если 1, то ошибка записи данных

Network 20: Title:

Передача данных на частотник, где  
E#DBX 8.0 BYTE 8 указатель на передачу 4 слов в DB301 начиная с адреса DBX 8.0

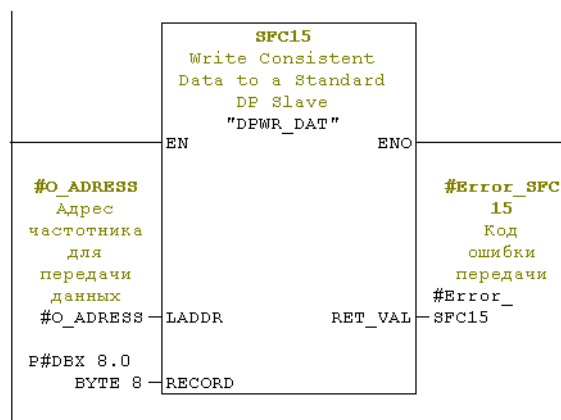


Рисунок 56 - Назначение параметров для блока передачи данных по сети Profibus SFC15  
Обрабатываем значение RET\_VAL SFC15, если =0 то статус состояния связи =:1

Network 21: Title:

Обратка статуса передачи данных на частотник

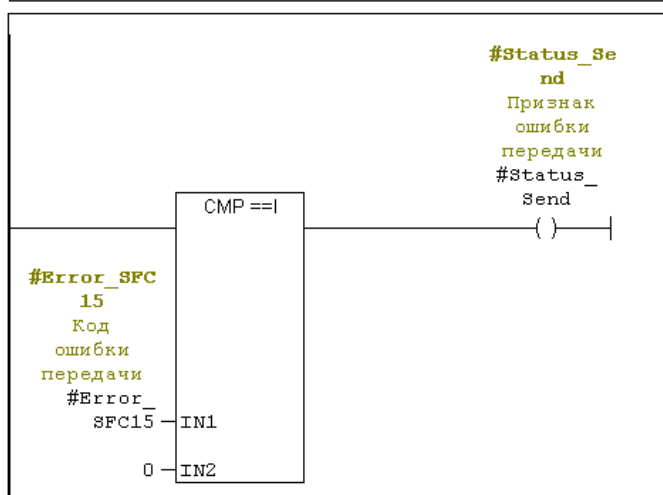


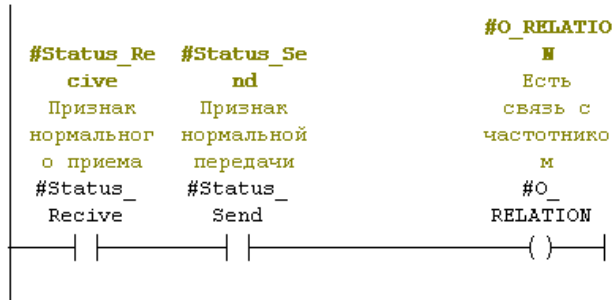
Рисунок 57 -Проверка отсутствия ошибок связи



Анализируем состояние связи с ПЧ, если статусы работы системных блоков SFC14/15 равны единицы, то выставляем на выходе сигнал «Есть связь с ПЧ»:

**Network 23** : Title:

Если при приеме и передаче данных нет ошибок, то выставляется признак - Есть связь!

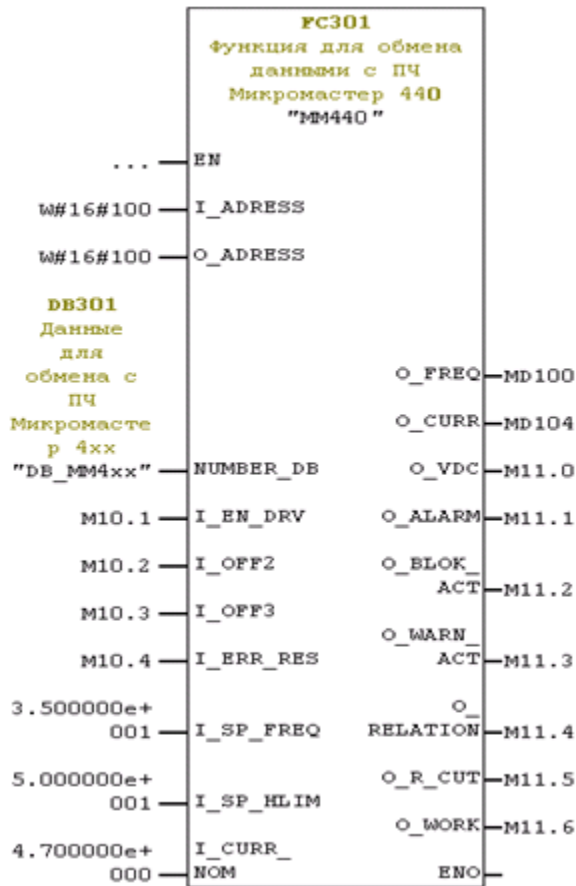


**Рисунок 58 - Проверка состояния связи на прием и передачу**

Организуем запуск функции FC301 из блока OB1. Для этого в организационный блок OB1 вставляем блок функцию FC301 и задаем входные и выходные переменные:

**Network 1** : Микромастер 440

В качестве примера использован параметр АД мощностью 0,37кВт и Iном=1,8А



**Рисунок 59 - Вставка блока FC301 в блок OB1**

Программа готова. Для включения электропривода в работу необходимо установить в 1 бит памяти M10.1, и задать уставку на вход функции "I\_SP\_FREQ".

Осуществляем итоговую настройку параметров ПЧ:

P0700=:6	Выбор источника команд;
P0840=:2090.0	Пуск/Стоп; Останов
P0844=:2090.1	выбегом;
P0845=:19.1	Останов выбегом от кнопки панели ВОР/ВОР2;
P0848=:2090.2	Быстрый останов;
P0852=:2090.3	Включение операционного режима;
P1035=:2090.13	Потенциометр вверх;
P1036=:2090.14	Потенциометр вниз;
P1055=:2090.8	Jog вправо (Не используется);
P1056=:2090.9	Jog влево (Не используется);
P1113=:2090.11	Реверс;
P1140=:2090.4	Активация наращивания уставки;
P1141=:2090.5	Фиксация заданной уставки;
P1142=:2090.6	Активация уставки;
P2103=:722.2	1 источник квитирования сбоя;
P2104=:2090.7	2 источник квитирования сбоя;
P2235=:2090.13	источник PID-МОР вверх;
P2236=:2090.14	источник PID-МОР вверх;
P0703=:99	ВICO дискретного входа 3;
P1000=:66	Выбор источника уставок частоты;
P1070=:2050.1	Главная уставка HSW;
P1075=:2050.1	Дополнительная уставка.

## Литература

1. Микромастер 440. Руководство пользователя. Издание А1. (6SE6400-5AA00-0AP00).
2. MICROMASTER. Дополнительный модуль связи PROFIBUS. Инструкция по эксплуатации Издание А1.(6SE6400-5AA00-0AP00).
3. Терехов В. М. Системы управления электроприводов : учебник /В. М. Терехов, О. И. Осипов ; под ред. В. М. Терехова. М.: Академия, 2005. 304 с.
4. Соколовский Г. Г. Электроприводы переменного тока с частотным регулированием: учебник / Г. Г. Соколовский. М.: Академия, 2006. 272 с.
5. Белов М. П. Автоматизированный электропривод типовых производственных механизмов и технологических комплексов: учебник для вузов/ М. П. Белов, В. А. Новиков, Л. Н. Рассудов. 2-е изд., стер. М.: Академия, 2004. 576 с.
6. Инжиниринг электроприводов и систем автоматизации: учебное пособие / М. П. Белов [и др.]; под ред. В. А. Новикова, Л. М. Чернигова. М.: Академия, 2006. 368 с.
7. Ю. Н. Федоров. Справочник инженера по АСУТП. Проектирование и разработка/ Ю. Н. Федоров. М.: Инфра-Инженерия, 2008. 927 с.