

Табличные функции	3.18
Техника доступа к параметрам в D7-SYS	3.19
Шина PROFIBUS DP	5

Коммуникация с WinCC (SINEC H1).

Проектирование коммуникаций D7-SYS

Издание 02/2002

Указания по технике безопасности

Данное руководство содержит указания, которые вы должны соблюдать для обеспечения собственной безопасности, а также защиты от повреждений продукта и связанного с ним оборудования. Эти замечания выделены предупреждающим треугольником и представлены, в соответствии с уровнем опасности следующим образом:



Опасность

указывает, что если не будут приняты надлежащие меры предосторожности, то это **приведет к гибели людей, тяжким телесным повреждениям или существенному имущественному ущербу.**



Предупреждение

указывает, что при отсутствии надлежащих мер предосторожности это **может привести к гибели людей, тяжким телесным повреждениям или к существенному имущественному ущербу.**



Осторожно

указывает, что возможны легкие телесные повреждения и нанесение небольшого имущественного ущерба при непринятии надлежащих мер предосторожности.

Осторожно

указывает, что возможно повреждение имущества, если не будут приняты надлежащие меры безопасности.

Замечание

привлекает ваше внимание к особо важной информации о продукте, обращении с ним или к соответствующей части документации.

Квалифицированный персонал

К монтажу и работе на этом оборудовании должен допускаться только **квалифицированный персонал**. Квалифицированный персонал – это люди, которые имеют право вводить в действие, заземлять и маркировать электрические цепи, оборудование и системы в соответствии со стандартами техники безопасности.

Надлежащее использование

Примите во внимание следующее:



Предупреждение

Это устройство и его компоненты могут использоваться только для целей, описанных в каталоге или технической документации, и в соединении только с теми устройствами или компонентами других производителей, которые были одобрены или рекомендованы фирмой Siemens.

Этот продукт может правильно и надежно функционировать только в том случае, если он правильно транспортируется, хранится, устанавливается и монтируется, а также эксплуатируется и обслуживается в соответствии с рекомендациями.

Товарные знаки

SIMATIC®, SIMATIC HMI® и SIMATIC NET® - это зарегистрированные товарные знаки SIEMENS AG.

Некоторые другие обозначения, использованные в этих документах, также являются зарегистрированными товарными знаками; права собственности могут быть нарушены, если они используются третьей стороной для своих собственных целей.

Copyright © Siemens AG 2002 Все права защищены

Воспроизведение, передача или использование этого документа или его содержания не разрешаются без специального письменного разрешения. Нарушители будут нести ответственность за нанесенный ущерб. Все права, включая права, вытекающие из патента или регистрации практической модели или конструкции, сохраняются.

Siemens AG

Департамент автоматизации и приводов
Промышленные системы автоматизации
Пля 4848, D- 90327, Нюрнберг

Siemens Aktiengesellschaft

Отказ от ответственности

Мы проверили содержание этого руководства на соответствие с описанным аппаратным и программным обеспечением. Так как отклонения не могут быть полностью исключены, то мы не можем гарантировать полного соответствия. Однако данные, приведенные в этом руководстве, регулярно пересматриваются, и все необходимые исправления вносятся в последующие издания. Мы будем благодарны за предложения по улучшению содержания.

©Siemens AG 2002
Technical data subject to change.



3.18 Табличные функции

3.18.1 Введение

Табличные функции SIMATIC TDC/SIMADYN D дают пользователю возможность применения при проектировании взаимосвязанных табличных величин. Для этого со стороны SIMATIC TDC или SIMADYN D должны быть спроектированы функциональные модули TAB и TAB_D. Функциональный модуль TAB оперирует табличными значениями формата REAL, TAB_D – табличными значениями формата DINT. Таким образом табличные величины становятся доступны пользователю.

Табличные функции могут быть спроектированы для трех способов пользования:

- **"Ручное" пользование:** означает, что табличные величины будут задаваться на модуль непосредственно по on-line интерфейсу (например, из CFC в режиме теста) и передаваться от модуля по специальному запросу программы (см. рис. 3-67).
- **Автоматическое пользование: коммуникация,** означает, что табличные величины будут передаваться по коммуникационному интерфейсу (TCP/IP, DUST1, S7 по Р-шине). Если табличные величины передаются по Р-шине от S7-процессора на прикладной модуль SIMATIC FM458, со стороны S7-процессора дополнительно проектируется функциональный модуль WR_TAB (см. рис. 3-68).
- **Автоматическое пользование: карточка памяти,** означает, что табличные величины будут загружены в карточку памяти и считываться оттуда.

ЗАМЕЧАНИЕ

Следует принимать во внимание, что изменение способов пользования возможно лишь между типами **"Ручное" пользование и Автоматическое пользование: коммуникация**, а также между **"Ручное" пользование и Автоматическое пользование: карточка памяти**

Задание или передача табличной величины сопровождается проверкой на корректность. Адрес таблицы индицируется на коннекторе "TAB".

Табличные значения дублируются, т. е. "управляются" двумя таблицами. Для всех без исключения вычислительных операций проекта используется таблица, определенная как "действительная" (=активная). "Недействительная" (=пассивная) таблица предназначена для контроля над изменениями величин. Все без исключения пользовательские изменения табличных величин воспринимаются сначала "недействительной" таблицей. Как только пассивная таблица будет активирована, новые табличные значения перепишутся (отразятся) во второй таблице. Активная до этого момента таблица теряет свою "действительность". Теперь новые табличные значения предоставлены к использованию в обеих таблицах.

Для предотвращения потери данных обе таблицы могут размещаться в сохраняемой области энергонезависимого ОЗУ (коннектор SAV=1 при инициализации).

ПРИМЕЧАНИЕ

Точное описание функциональных модулей TAB и TAB_D вы найдете в соответствующей Online- подсказке.

Точное описание функционального модуля WR_TAB вы найдете в следующей ниже главе "Функциональный модуль WR_TAB".

3.18.1.1 Обзор способа пользования "ручное".

Следующая иллюстрация показывает принцип работы при "ручном" пользовании:

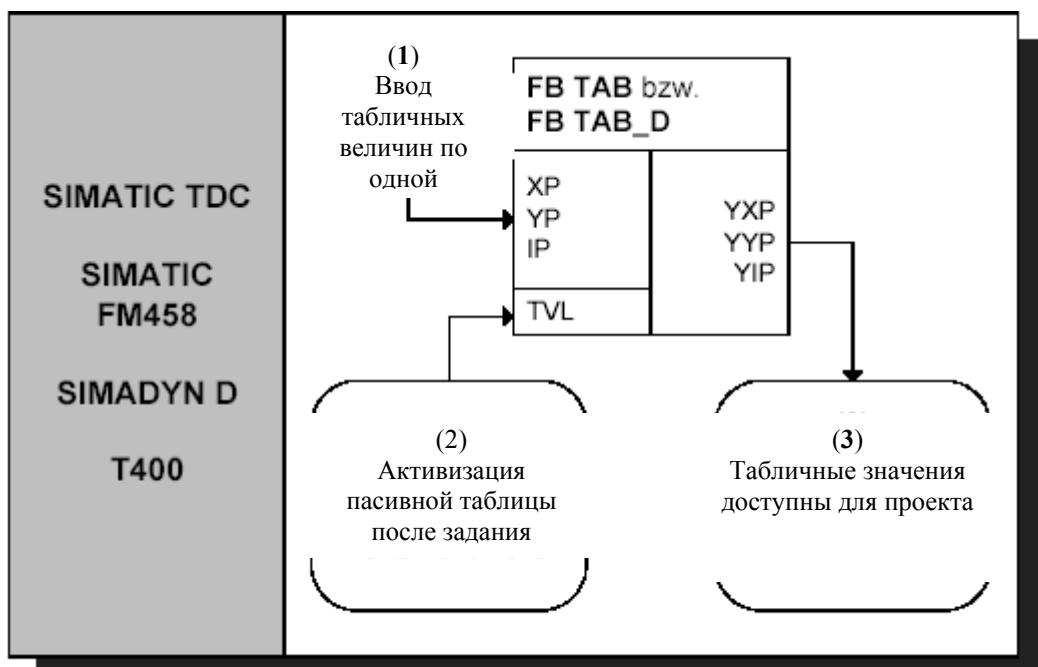


Рис. 3-67 Принцип работы в режиме "ручного" пользования

Подробное описание режима "ручное" пользование смотри раздел "Ручное пользование" (стр. 7).

3.18.1.2 Обзор режима "Автоматическое пользование: коммуникации".

В режиме "Автоматическое пользование: коммуникации" табличные значения могут передаваться при помощи следующих вариантов коммуникаций:

- S7 по Р-шине для SIMATIC FM 458 (со стороны S7-задачи требуется дополнительное проектирование WR_TAB)
- TCP/IP (табличные значения посредством функций CTV и CRV могут передаваться также и от одного модуля SIMATIC TDC к другим)
- DUST1 (табличные значения могут передаваться по интерфейсу DUST1)

Табличные величины могут передаваться в телеграмме.

Следующая иллюстрация показывает принципиальные положения для режима "Автоматическая работа: коммуникации" в случае передачи таблицы от S7-задачи к прикладному модулю SIMATIC FM 458 по Р-шине:

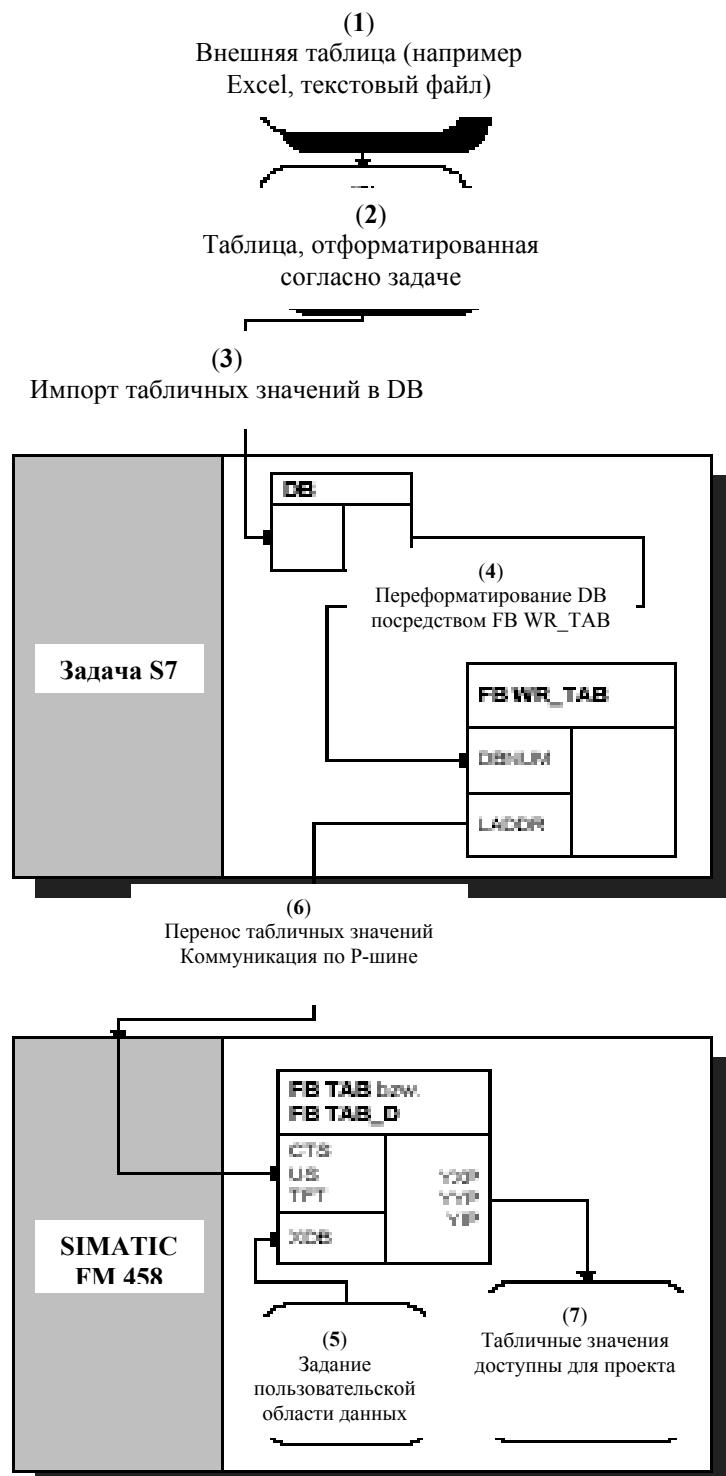


Рис. 3-68 Принципиальные положения режима "Автоматическое пользование: коммуникации (на примере коммуникации по Р-шине).

Подробное описание режима "Автоматическая работа: коммуникации" для передачи таблицы от S7-задачи на прикладной модуль SIMATIC FM 458 смотри главу "Автоматическая работа: коммуникации" (стр. 9).

3.18.1.3 Обзор режима "Автоматическое пользование: карточка памяти".

Постедством режима "Автоматическое пользование: карточка памяти" можно загрузить табличные значения в карточку памяти (посредством *D7-SYS additionalComponentBuilder*). При инициализации модуля эти данные будут активированы.

Подробное описание режима "Автоматическая работа: карточка памяти" смотри главу "Автоматическая работа: карточка памяти" (стр. 30).

3.18.1.4 Функциональный модуль WR_TAB.

Символ

WR_TAB			
Активизация модуля	- BO	EN	TABTEL
Требование записи новой таблицы	- BO	REQTAB	CNTTEL
Требование записи табличных значений из блока данных DB	- BO	REQDB	STATUS
Последний блок данных для таблицы Логический адрес модуля (FM 458)	- BO	LASTDB	ERROR
	- W	LADDR	DONE
Номер набора данных для считывания /записи набора	- BY	RECNUM	
Номер блока данных TIME-OUT (на передачу подтверждения от FM 458)	- W	DBNUM	
	- DW	TFT	

Число блоков данных, потребных для передачи полного содержимого всех DB (таблицы)

Число уже переданных блоков данных

Текущий статус работы

Код сбоя

Параметр состояния DONE: процесс передачи прерван

Краткое описание

Функциональный модуль WR_TAB предназначен для передачи таблиц от задач S7 на прикладной модуль SIMATIC FM 458. Табличные величины (допустимые типы данных REAL и Double Integer) задаются блоком данных (DB). Посредством WR_TAB эти табличные величины передаются функциональным модулям TAB или TAB_D прикладного модуля SIMATIC FM 458, которые определяют табличные значения внутри программы FM 458.

Функциональный модуль WR_TAB проектируется со стороны генерации таблицы. Передача табличных значений осуществляется от процессора S7-400 по Р-шине к прикладному модулю SIMATIC FM 458. Причем всегда передаются все значения, принадлежащие блоку данных (DB), номер которого указан на "входе" DBNUM.

Коннекторы

Параметр	Декларация	Тип данных	Описание
REQTAB	INPUT	BOOL	REQTAB=1; Требование записи новой таблицы
REQDB	INPUT	BOOL	REQDB=1; требование записи табличных значений, выделенных из блока данных
LASTDB	INPUT	BOOL	Последний DB для таблицы
LADDR	INPUT	WORD	Логический адрес прикладнлгл модуля SIMATIC FM 458
RECNUM	INPUT	BYTE	Номер набора данных для считывания/записи набора
DBNUM	INPUT	WORD	Номер блока данных, DB, из значений которого состоит таблица
TFT	INPUT	DWORD	TIME-OUT, время ожидания поступления подтверждающей телеграммы от FM458, в mS
TABTEL	OUTPUT	WORD	Число блоков данных, потребных для передачи полного содержимого всех DB (таблицы)
CNTTEL	OUTPUT	WORD	Число уже переданных на FM блоков данных
STATUS	OUTPUT	WORD	Указатель текущего статуса обработки/передачи: 0: передача таблицы деактивирована 1: передача таблицы активна. Производится передача табличных значений из блока данных DB. (Ожидание следующего цикла передачи). 2: передача табличных значений из текущего блока данных не завершена
ERROR	OUTPUT	WORD	Если при обработке функции возникает ошибка, возвращаемое значение будет содержать код ошибки
DONE	OUTPUT	BYTE	Параметр состояния DONE=1: процесс передачи прерван

Могут быть зафиксированы и индицированы на выходе ERROR следующие ошибки:

Код сбоя	Пояснение	Рекомендация
0xB210	OK	-
0xB211	Некорректный логический адрес модуля (FM 458)	Задайте правильный адрес модуля (FM 458) на входе LADDR
0xB212	Неверен номер набора данных	Табличные значения необходимо заносить в DB в восходящей последовательности ¹
0xB213	Неверный формат данных в таблице	Табличные величины должны иметь формат REAL для TAB и DINT для TAB D
0xB214	Формат данных нового набора данных не соответствует формату до сих пор передаваемым наборам	Обратите внимание на то, что табличные значения должны иметь идентичный формат
0xB215	FM 458 не отвечает	Проверьте проект и коммуникационную связь
0xB216	Таблица велика	Передавайте таблицу частями, т. е. либо разделите табличные значения на большее число DB, либо для каждого цикла передачи записывайте в DB новые (следующие) табличные значения.
0xB217	Таблица не заполнена (значения X- / Y-)	Проверьте заполнение таблицы, каждому значению X должно быть поставлено в соответствие значение Y.
0xB218	REQTAB сбрасывается во время обработки	Вновь проведите передачу табличных значений
0xB219	REQDB сбрасывается во время обработки	Вновь проведите передачу табличных значений
0xB21A	Номер DB неверен	Задайте правильный номер DB.
0xB21B	TIME-OUT при приеме подтверждающей телеграмме	Проверьте проект и коммуникационную связь. Повторите передачу табличных значений.
0xB21C	Недопустимое рабочее состояние	Проверьте проектирование WR TAB

¹ Имеется в виду последовательное, в порядке возрастания НОМЕРА (адреса), задание пары "аргумент-функция", т.е. x1, y1; x2, y2; x3, y3; но не x1, y1; x3, y3; x2, y2.

3.18.2 Ручное пользование

3.18.2.1 Применение

Способ пользования "ручное" представляет наиболее простую возможность введения в проект табличных величин. Однако ручной ввод данных и, соответственно, их вызов Teach-in, требует сравнительно много времени.

Ввод табличных значений

После корректного проектирования TAB или TAB_D можно последовательно, друг за другом задавать табличные значения. Прежде всего, на вход NP задается размер таблицы, т.е. число пар величин (= точек) таблицы. Если таблица должна запоминаться в области энергонезависимой памяти (SAVE) вход SAV функционального модуля должен быть установлен в 1.

Затем могут быть заданы значения табличных величин. Первоначально на входе IP указывается индекс i задаваемой пары значений. После этого на входы XP и YP задаются значения X и Y. Для того чтобы заданные значения были восприняты, ввод каждой пары величин необходимо завершать установкой входа WR из 0 в 1. Перед вводом значений следующей точки таблицы необходимо переустановить значение индекса на входе IP. Теперь задается значение новой точки таблицы. Эта процедура повторяется до тех пор, пока не будут введены все табличные значения.

Число введенных точек таблицы должно соответствовать значению, заданному на входе NP.

Все задания во время проведения этой процедуры воспринимаются пассивной таблицей функционального модуля и будут доступны в проекте только после активизации. Для активизации пассивной таблицы с введенными табличными величинами необходимо установить 1 на входе TVL.

Последующие изменения будут заноситься в пассивную таблицу и предоставляться к пользованию после повторной активизации.

Опрос табличных значений

Для вызова табличного значения необходимо, после завершения ввода, задать на входе IP индекс опрашиваемой точки и установить вход RD из 0 в 1. Табличные значения точки будут доступны на выходах YXP (значение X) и YYP (значение Y). Индекс точки i будет выдан на выход YIP.

3.18.2.2 Проектирование

Для использования в режиме "ручное" могут быть спроектированы только TAB и/или TAB_D, в зависимости от того, данные какого типа (REAL или DINT) представлены в таблице. Каждая таблица должна содержать значения одного типа данных. Если имеется несколько таблиц с различными типами данных – для каждой таблицы должен быть спроектирован свой TAB или TAB_D.

Функциональные модули TAB и TAB_D должны проектироваться в задаче с временем опроса большим или равным 32 mS. Обязательны следующие установки на коннекторах:

AUT = 0 (деактивирована автоматическая работа)
NP = [задание размера таблицы]
XP = [задание значения X]
YP = [задание значения Y]
IP = [задание индекса изменяемой пары значений]
TVL = 1 (активизация таблицы после задания всех значений)
WR = 1 (для записи пары значений в таблицу)
RD = 1 (для считывания на выхода YXP и YYP пары значений, указанных на IP)

В случае, когда в режиме "ручное" коннектор CTS был при инициализации установлен в "0" (CTS=0; AUT=0), переключение на режим "автоматическое пользование: карточка памяти" (CTS=0; AUT=1) становится невозможным.

В случае, когда при инициализации коннектор CTS был установлен в "0" и был активирован режим "автоматическое пользование: карточка памяти" (AUT=1), появляется возможность переключения на режим "ручное" (CTS=0; AUT=0). Запомненная в карточке памяти таблица будет обрабатываться в режиме "ручное".

Попытка возврата к режиму работы "автоматическое пользование: карточка памяти" (CTS=0; AUT=1) не будет удовлетворена, т. к. это возможно лишь в процессе инициализации.

При проектировании на коннекторе CTS коммуникационного интерфейса возможны любые переключения между режимами работы "ручное" и "автоматическое пользование: коммуникация"

3.18.3 Автоматическое пользование: коммуникация

3.18.3.1 Применение для S7-задачи и прикладного модуля SIMATIC FM 458

Передача табличных значений

Для успешной передачи таблицы должны быть соблюдены следующие условия:

- На прикладном модуле FM458 должны быть спроектированы функциональные модули TAB и/или TAB_D, соответствующие режиму "автоматическое пользование: коммуникации". (Подробные разъяснения по этому поводу Вы найдете в следующем разделе "Проектирование для S7-задачи и прикладного модуля SIMATIC FM 458").
- Значения X и Y в содержащейся в DB таблице должны быть записаны попарно. Для каждого значения X должно существовать значение Y, так что число значений должно быть всегда четным.

Для того, чтобы начать процесс передачи, нужно установить в "1" входы REQTAB и REQDB модуля WR_TAB. После этого начнется передача табличных значений из DB, номер которого указан на DBNUM модуля WR_TAB.

На выходе CNTTEL модуля WR_TAB при этом всегда индицируется текущее число переданных блоков данных.

На выходе TABTEL модуля WR_TAB будет индицировано число блоков данных, которое потребовалось для переноса на прикладной модуль SIMATIC FM 458 полного содержимого всех DB (таблицы).

Если все значения таблицы полностью находятся в указанном DB, или если речь идет о последней части передачи таблицы в случае, когда один DB не соответствует полному содержанию таблицы; до начала этой передачи вход LASTDB модуля WR_TAB нужно установить в "1". Одновременно прикладной модуль SIMATIC FM 458 сигнализирует о завершении передачи таблицы. Выход STATUS модуля WR_TAB изменяется с 2 на 0.

ЗАМЕЧАНИЕ

Всегда передаются все табличные значения, содержащиеся в DB, номер которого задан на входе DBNUM модуля WR_TAB.

Таблица велика для одного DB

В том случае, когда таблица велика для одного блока данных (DB), нужно разделить передачу табличных значений на части. При этом необходимо поступать следующим образом:

Вначале необходимо записать первую часть таблицы в DB и передать, как описано выше. Вход LASTDB модуля WR_TAB остается в 0. Выход STATUS модуля WR_TAB устанавливается во время передачи на 2 и, по окончании первой части передачи, переходит от 2 к 1.

Затем старые табличные значения DB переписываются последующими значениями. Когда это выполнено, вход REQDB модуля WR_TAB обновляется установкой из 0 в 1 для активизации передачи следующей части таблицы.

Эта процедура повторяется до тех пор, пока не будут переданы все табличные значения.

При передаче последней части таблицы вход LASTDB модуля WR_TAB нужно установить из 0 в 1. При этом прикладной модуль SIMATIC FM 458 сигнализирует о завершении передачи таблицы. Выход STATUS модуля WR_TAB изменяется с 2 на 0

ЗАМЕЧАНИЕ

Если в распоряжении пользователя имеется достаточный объем памяти, таблица может быть представлена несколькими DB. В этом случае при каждой передаче части таблицы на вход DBNUM модуля WR_TAB задается номер соответствующего DB. При этом нужно обращать внимание на то, чтобы DB передавались в правильной последовательности, так, чтобы все табличные значения были переданы в порядке возрастания индекса.

Длительность передачи

Время передачи табличных значений зависит от следующих факторов:

- число табличных значений
- размер блока данных
- Время опроса TAB или TAB_D
- Время обработки WR_TAB

В каждом цикле от S7-процессора к прикладному модулю SIMATIC FM 458 переносится телеграмма с 56-ю значениями.

Время передачи таблицы можно рассчитать следующим образом:

Время передачи = [число табличных значений/56] * время цикла самого "медленного" FB (т. е. TAB, TAB_D или WR_TAB).

Время передачи данных по Р-шине для этой оценки не имеет значения, т. к. этот период времени как правило менее 1mS, а функциональные модули TAB или TAB_D как правило проектируются в задаче с временем опроса более 32mS.

Если таблица разделена на несколько блоков данных, период времени передачи повышается. Это происходит потому, что дополнительно к времени передачи таблицы, которое может быть определено по приведенной выше формуле, необходимы описанные выше "ручные" пользовательские изменения.

3.18.3.2 Проектирование для S7-задачи и прикладного модуля SIMATIC FM 458

Для связи S7-задачи с прикладным модулем SIMATIC FM 458 по Р-шине необходимо спроектировать следующие функциональные модули:

- Прикладной модуль SIMATIC FM 458:
 - TAB (для данных типа REAL) и/или
 - TAB_D (данные типа DINT)
 - @CBP (центральный модуль связи по Р-шине)
- S7-задача:
 - WR_TAB

Каждая таблица должна содержать значения одного типа данных. Если имеется несколько таблиц, содержащих данные различного типа, для каждой таблицы должен быть спроектирован соответственно TAB или TAB_D.

WR_TAB предназначен для передачи табличных значений от SIMATIC DB к функциональным модулям TAB или TAB_D. Табличные значения переносятся в телеграмме, содержащей данные таблицы. В последнюю из передаваемых для модулей TAB или TAB_D телеграмм втоматически добавляется информация о том, что все табличные значения были переданы; таблица активируется. WR_TAB получает подтверждающее сообщение о том, успешно или нет завершена активизация. В случае успешной активизации таблицы ее адрес будет вынесен на выход TAB модуля TAB или TAB_D.

ТАВ или TAB_D.

ТАВ или TAB_D проектируются следующим образом:

Модули должны быть спроектированы в задаче со временем опроса большим или равным 32mS. Обязательны следующие объявления на коннекторах:

CTS = [Имя проектируемого коммуникационного интерфейса]
AUT = 1 (активирована автоматическая работа)
US = [Имя канала.Первая часть адреса] (Задание адреса для приема)
MOD = [Режим связи] (H=Handshake; R=Refresh; S=Select; M=Multiple)
TFT = [Контрольное время в mS] (максимально допустимое время "выпадания" телеграммы при приеме табличных значений)
NP = [Задание максимального размера таблицы]

ЗАМЕЧАНИЕ Если на коннекторе CTS спроектирован коммуникационный интерфейс, становятся возможными любые переключения между режимами "автоматическое пользование: коммуникации" и "ручное".

WR_TAB.

На коннекторах WR_TAB проецируются следующие объявления¹:

LADDR = [Задание логического адреса прикладного модуля SIMATIC FM 458]
RENUM = [Задание номера набора данных для канала записи/считывания. Должно быть идентично заданию "Первой части адреса" на коннекторе US модуля TAB или TAB_D.]
DBNUM = [Задание номера блока данных]

3.18.3.3 Ввод табличных значений в блок данных

Для передачи табличных значений на прикладной модуль SIMATIC FM 458 необходим блок данных (DB). DB программируется на стороне S7-задачи.

Имеется два способа создания DB с желаемыми табличными значениями:

- Создание нового DB в STEP7 и задание табличных значений в приложении "KOP/FUP/AWL" вручную
- Импорт табличных значений из уже созданной таблицы (например, MS Excel) в STEP7

3.18.3.3.1 Ввод табличных значений вручную

Здесь речь идет о наиболее простом методе предоставления табличных значений для DB. Он состоит в том, что начальное/актуальное значение каждой табличной величины вводится, посредством приложения "KOP/FUP/AWL", во вновь созданный DB вручную. Требуемая для этого метода последовательность действий будет описана в дальнейшем.

ЗАМЕЧАНИЕ Начальное значение для каждой табличной величины может быть задано произвольно. Оно будет использовано тогда, когда для соответствующего табличного значения не определена актуальная величина.

Актуальное значение – это значение, которое представлено в проекте в качестве табличного значения. Здесь задается желаемое табличное значение.

¹ Для функции WR_TAB необходимо продекларировать следующие входные переменные – но это будет уже не дословный перевод. А дословный перевод "нагибает" в сторону рисования схем. Может это и правильно.

(1) Создание нового DB в STEP7

Первым делом необходимо создать новый DB в STEP7. Для этого необходимо отметить папку "Bausteine" ("Модули") в соответствующей S7-программе и выбрать в контекстном меню элемент "Neues Objekt einfügen → Datenbaustein" ("Добавить новый объект → Блок данных").

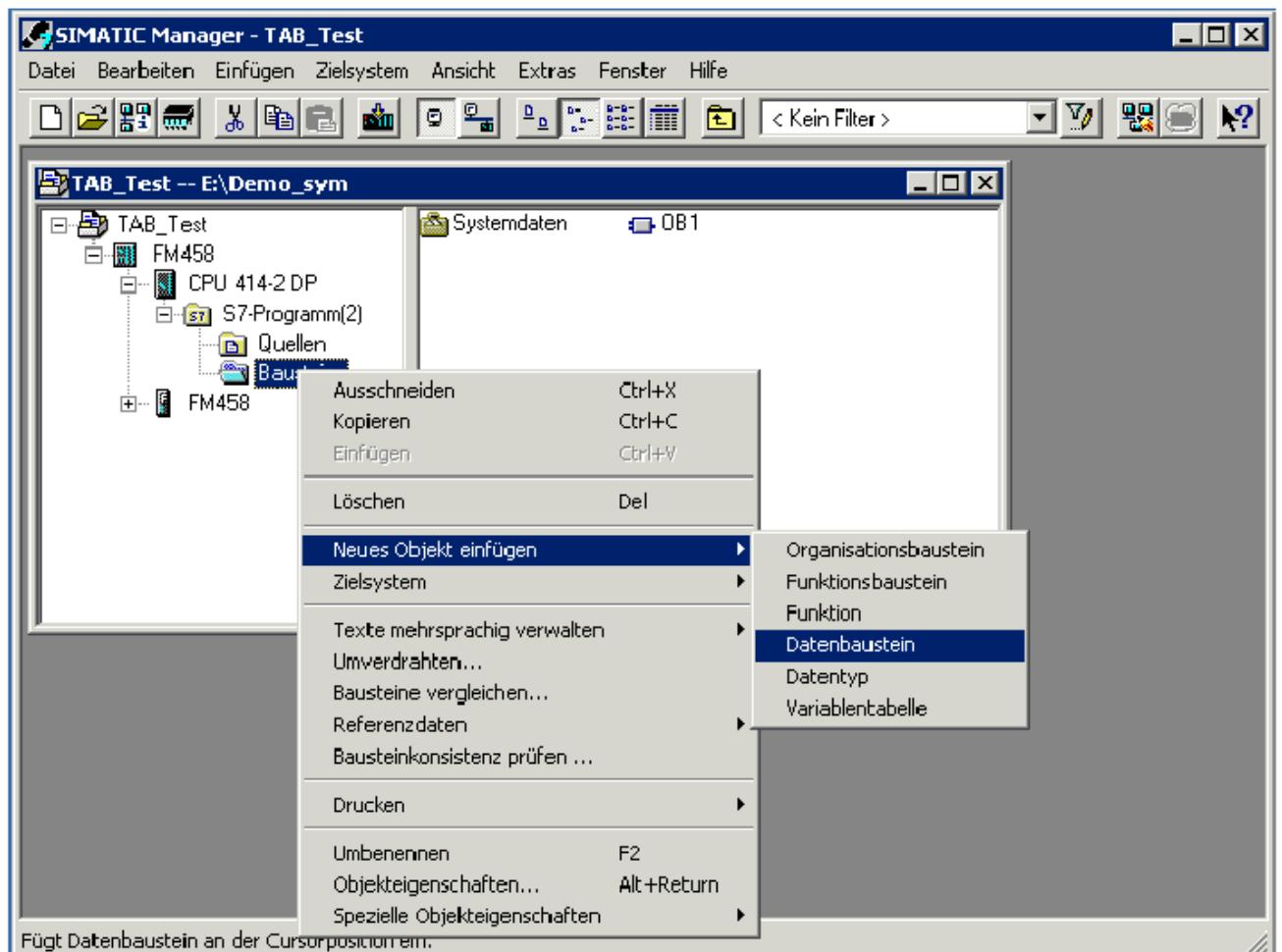


Рис. 3-69. Создание нового блока данных в STEP7.

(2) Открытие нового DB

Следующий шаг состоит в открытии вновь созданного DB в приложении "KOP/FUP/AWL", что производится двойным щелчком мыши по DB. Инструментарием (**Erstellungswerkzeug**) является "Редактор DB" ("DB-Editor") и создается (**Erstellen**) только блок данных (**Datenbaustein**).

На следующем рисунке показан выбор (пунктов меню) при открытии нового DB:

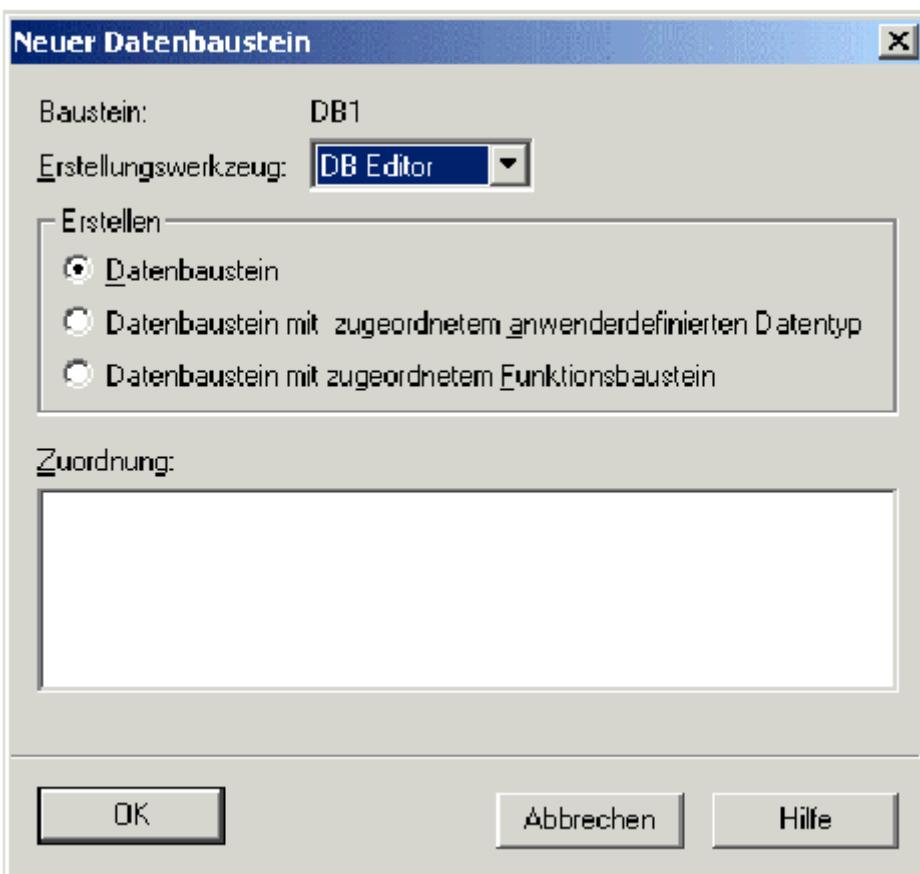


Рис. 3-70. Выбор (пунктов меню) при создании нового DB.

Следующий рисунок показывает открытый новый DB:

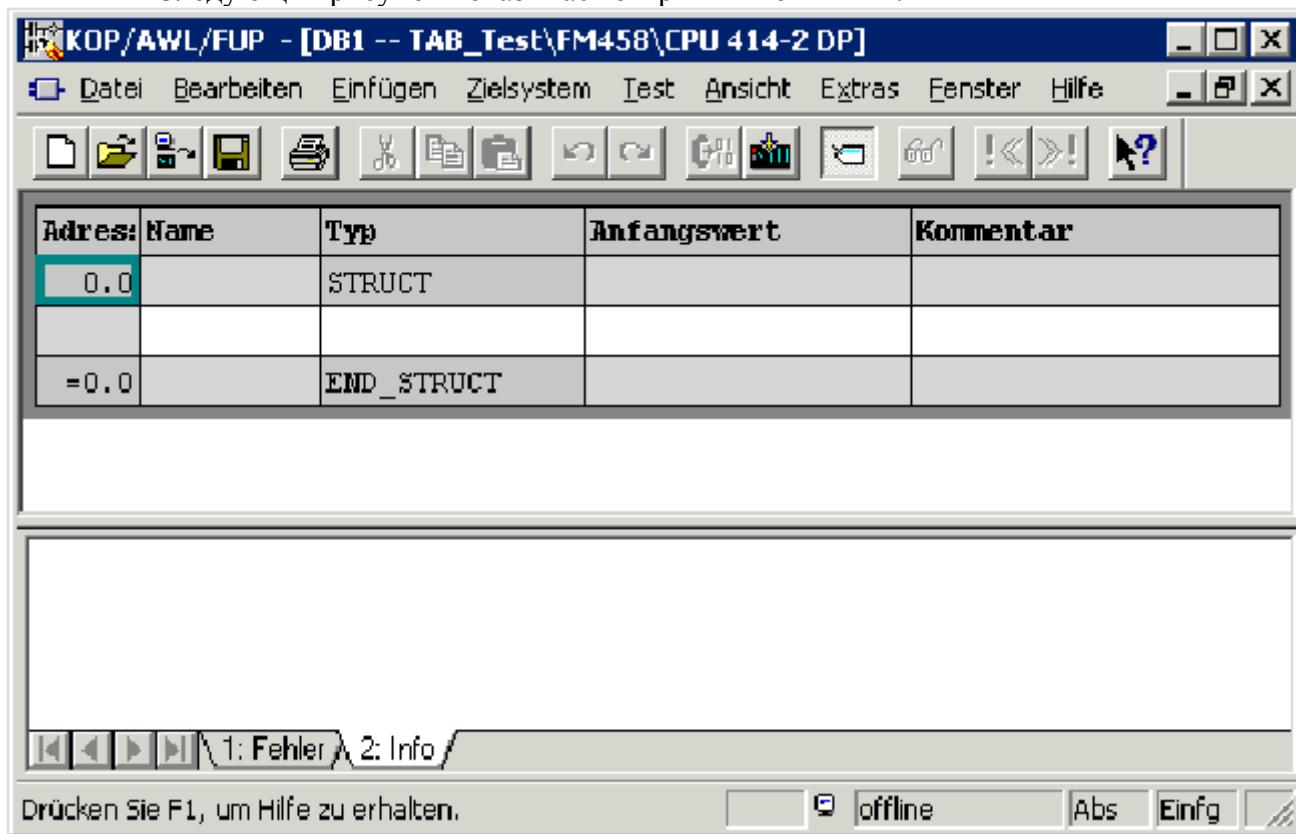


Рис. 3-71. Вновь созданный в приложении "KOP/FUP/AWL" DB.

(3) Задание табличных значений

Должны быть заданы только желаемые табличные значения. При этом следует обращать внимание на то, чтобы значения X и Y задавались попеременно.

Вначале задается тип данных для таблицы. При этом всегда в качестве имени (Name) задается "Datatype", тип (Type) – "WORD" и начальное значение для данных типа REAL (Anfangswert) – "W#16#1", данных типа DINT - "W#16#2". Затем для каждой табличной величины задаются соответствующие имя, тип данных (колонка "Typ") и значение (колонка "Anfangswert").

Следующий рисунок иллюстрирует принцип действия при задании табличных значений данных типа REAL:

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	Datatype	WORD	W#16#1	
+2.0	x1	REAL	1.000000e+000	
+6.0	y1	REAL	5.000000e+000	
+10.0	x2	REAL	2.000000e+000	
+14.0	y2	REAL	6.000000e+000	
+18.0	x3	REAL	3.000000e+000	
+22.0	y3	REAL	7.000000e+000	
+26.0	x4	REAL	4.000000e+000	
+30.0	y4	REAL	8.000000e+000	
=34.0		END_STRUCT		

Рис. 3-72. Ручное задание табличных значений в приложении "KOP/FUP/AWL".

ЗАМЕЧАНИЕ Так как одной таблице могут принадлежать только данные одного и того же типа, эффективным способом задания является определение области (ARRAY). В этом случае нет необходимости задавать тип данных для каждой величины.

Образ действия при создании объявлений типа ARRAY смотри on-line помощь приложения "KOP/FUP/AWL", в особенности "помощь в AWL".

(4) Сохранение DB

По завершении задания табличных значений DB должен быть сохранен посредством "**Datei → Speichern**" ("Файл → Сохранить").

Табличные значения DB готовы к передаче.

3.18.3.3.2 Импорт табличных значений

Табличные значения, поставленные в соответствие данному DB, могут быть также импортированы из внешнего источника, например, из таблицы MS Excel.

Для безошибочного импорта обращайте внимание на следующие положения:

- Исходный файл таблицы должен иметь определенный формат
- Исходный файл должен быть встроен в STEP7 как внешний исходный файл
- Новый DB создается посредством внешнего исходного файла

Необходимые для процесса импорта этапы и шаги разъясняются далее.

Формат таблицы

Для того, чтобы уже готовая таблица (например, созданная в MS Excel) могла быть импортирована в DB, эта таблица должна удовлетворять определенным требованиям с точки зрения формата:

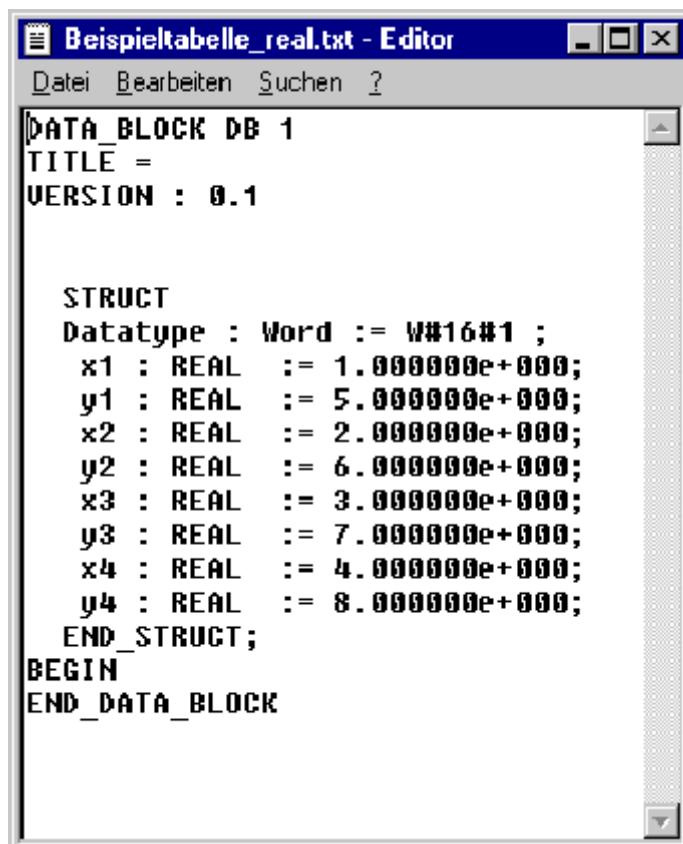
- Таблица должна иметь заголовок, содержащий информацию об имени DB и версии
- Далее должна быть объявлена структура и тип данных табличных величин
- Затем следуют задания табличных значений (начальные значения)
- При этом следует обращать внимание на то, что значения X и Y должны всегда чередоваться
- Таблица должна быть сохранена с расширением **AWL**
- Теперь таблица может быть использована как внешний исходный файл

ЗАМЕЧАНИЕ Начальное значение для каждой табличной величины может быть задано произвольно. Оно будет использовано тогда, когда для соответствующего табличного значения не определена актуальная величина.

Табличные значения определяются исключительно как *начальные значения*.
Актуальные значения не используются.

Вследствие этого существенно уменьшается размер файла и, соответственно, требуемый объем памяти.

Следующий рисунок показывает пример таблицы с четырьмя парами значений X и Y типа REAL:



```

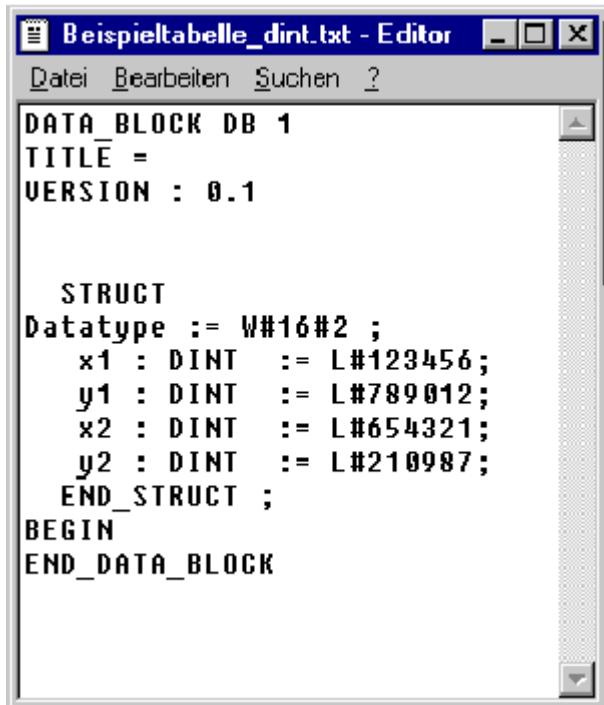
Beispieltablett_real.txt - Editor
Datei  Bearbeiten  Suchen  ?
DATA_BLOCK DB 1
TITLE =
VERSION : 0.1

STRUCT
Datatype : Word := W#16#1 ;
x1 : REAL    := 1.000000e+000;
y1 : REAL    := 5.000000e+000;
x2 : REAL    := 2.000000e+000;
y2 : REAL    := 6.000000e+000;
x3 : REAL    := 3.000000e+000;
y3 : REAL    := 7.000000e+000;
x4 : REAL    := 4.000000e+000;
y4 : REAL    := 8.000000e+000;
END_STRUCT;
BEGIN
END_DATA_BLOCK

```

Рис. 3-73. Пример таблицы со значениями типа *REAL*.

Следующий рисунок показывает пример таблицы с двумя парами значений X и Y типа DINT:



```
DATA_BLOCK DB 1
TITLE =
VERSION : 0.1

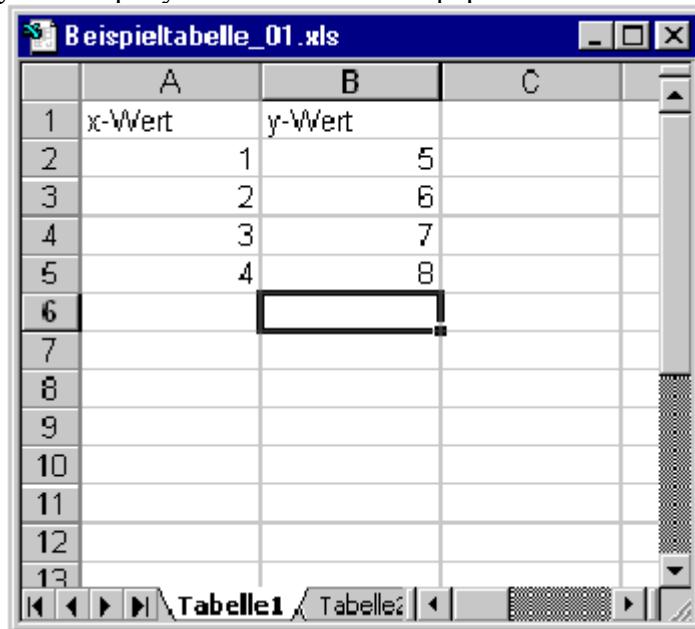
STRUCT
Datatype := W#16#2 ;
x1 : DINT    := L#123456;
y1 : DINT    := L#789012;
x2 : DINT    := L#654321;
y2 : DINT    := L#210987;
END_STRUCT ;
BEGIN
END_DATA_BLOCK
```

Рис. 3-74. Пример таблицы со значениями типа DINT.

От Excel к AWL

Текущий раздел поясняет на примере характер действий при преобразовании таблицы Excel в требуемый формат.

На следующих иллюстрациях будет показано шаг за шагом формирование файла для получения требуемого табличного формата:



	A	B	C
1	x-Wert	y-Wert	
2		1	5
3		2	6
4		3	7
5		4	8
6			
7			
8			
9			
10			
11			
12			
13			

Рис. 3-75. Пример таблицы MS Excel.

(1) Заголовок

Вначале добавляем требуемый заголовок. В начале (таблицы) добавляем 5 строк и вносим следующую информацию:

- DATA_BLOCK DB 1 [номер DB]
- TITLE = [задается при необходимости]
- VERSION: 0.1 [идентификатор версии]

Следующий рисунок показывает таблицу Excel с добавленным заголовком

Рис. 3-76. Пример таблицы MS Excel с добавленным заголовком.

(2) Добавления структуры и табличных значений

На следующем этапе добавляется структура табличных значений и собственно табличные значения с указанием типа данных. При этом для каждой пары значений добавляется пара строк, плюс строка в начале и строка в конце. Кроме того, для объявления типа данных в начале (таблицы) добавляется еще одна строка.

Начало объявления структуры отмечается занесением в начальную строку идентификатора "STRUCT". В следующей строке объявляется используемый в таблице тип данных **Datatype** ("W#16#1" для данных типа REAL, "W#16#2" для данных типа DINT).

Затем производится задание структуры и собственно табличных величин, попарно, значения X и Y всегда чередуются. Табличные величины задаются в соответствии с используемым типом данных (здесь – REAL). Завершение формирования структуры отмечается в заключительной строке идентификатором "END_STRUCT".

В заключение нужно лишь объявить часть данных для актуальных значений ("BEGIN" и "END_DATA_BLOCK"). Так как табличные величины уже определены в виде начальных значений заданием структуры, объявления отдельных актуальных значений может быть опущено.

Следующий рисунок показывает таблицу Excel с добавленными объявлениями структуры и табличными значениями:

	A	B	C
1	DATA_BLOCK DB 1		
2	TITLE =		
3	VERSION : 0.1		
4			
5			
6	STRUCT		
7	Datatype := W#16#1 ;		
8	x1 : REAL := 1.000000e+000;		
9	y1 : REAL := 5.000000e+000;		
10	x2 : REAL := 2.000000e+000;		
11	y2 : REAL := 6.000000e+000;		
12	x3 : REAL := 3.000000e+000;		
13	y3 : REAL := 7.000000e+000;		
14	x4 : REAL := 4.000000e+000;		
15	y4 : REAL := 8.000000e+000;		
16	END_STRUCT;		
17	BEGIN		
18	END DATA_BLOCK		
19			
20			
21			
22			

Рис. 3-77. Пример таблицы MS Excel с добавленными объявлениями структуры и табличными значениями.

(3) Сохранение в виде файла AWL

На заключительном этапе корректно отформатированный текстовый файл сохраняем в виде файла с расширением AWL. Для этого в файле MS Excel выбираем "Файл → Сохранить как...". Выбираем тип файла "Форматированный текст (разделители - пробелы) (*.prn)" и сохраняем таблицу под любым именем в любом каталоге.

Следующая картинка показывает окошко "Сохранить как..." в MS Excel и соответствующий выбор:

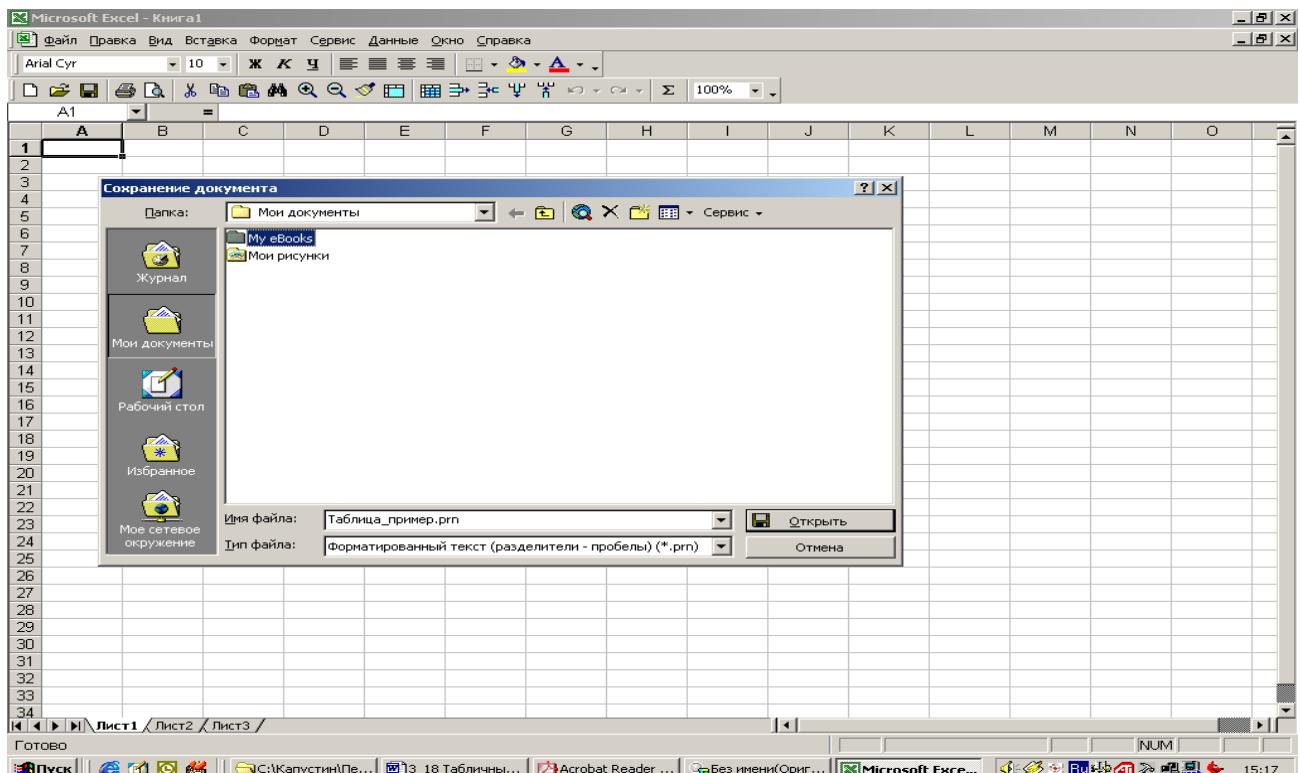


Рис. 3-78. Сохранение таблицы MS Excel в виде текстового файла (*.prn)

После сохранения файла изменяем его расширение с prn на awl. Теперь этот файл может быть открыт любым текстовым редактором.

На следующем рисунке показан пример таблицы в виде файла AWL (**beispieltabelle.awl**), открытый стандартным текстовым редактором Windows (Блокнот).

```

beispieltabelle.awl - Editor
Datei Bearbeiten Suchen ?
DATA_BLOCK DB 1
TITLE =
VERSION : 0.1

STRUCT
Datatype : Word := W#16#1 ;
x1 : REAL := 1.000000e+000;
y1 : REAL := 5.000000e+000;
x2 : REAL := 2.000000e+000;
y2 : REAL := 6.000000e+000;
x3 : REAL := 3.000000e+000;
y3 : REAL := 7.000000e+000;
x4 : REAL := 4.000000e+000;
y4 : REAL := 8.000000e+000;
END_STRUCT;
BEGIN
END_DATA_BLOCK

```

Рис. 3-79. Открытый в текстовом редакторе пример таблицы, сохраненный в виде файла AWL.

Этот файл может служить внешним исходным файлом для DB в STEP7.

Вставка таблицы (в STEP7) в качестве исходного файла

На основе созданного выше файла примера "Beispieltablette.awl" разъясняется последовательность действий для встраивания внешней таблицы в DB.

Кроме задания табличных значений особое внимание следует обращать также на задание имени DB. В зависимости от объявленного в файле имени позже будет создан новый DB.

В упомянутом выше файле примера в качестве имени DB в первой строке таблицы задано "DB1" (см. рис. 3-76).

При проектировании STEP7 в S7-программу в качестве "источников" ("Quellen") может быть выбран только один внешний источник (**Externe Quelle**). Вызов контекстного меню выбора источников осуществляется посредством "клика" правой клавишей мышки на правой правой части окна. Здесь внешний источник (**Externe Quelle**) добавляется как новый объект (**Neues Objekt einfügen**).

Следующий рисунок отображает последовательность действий:

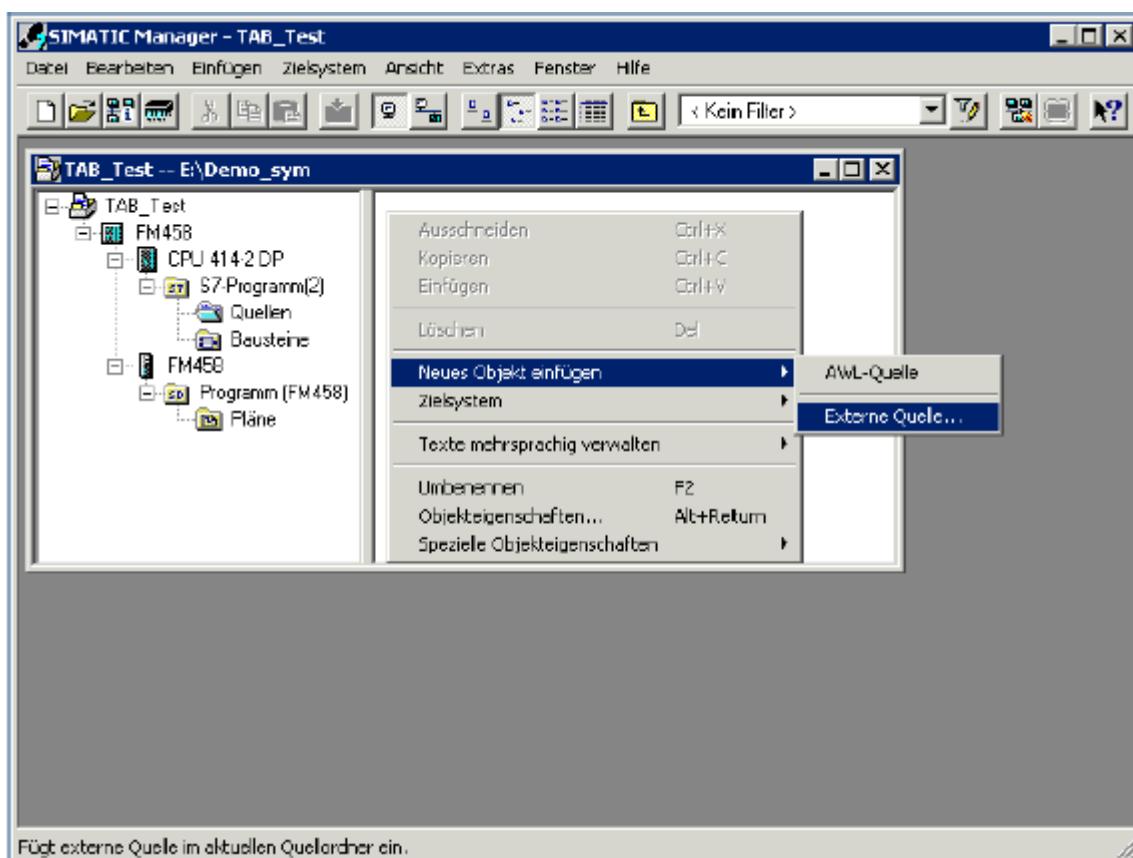


Рис. 3-80. Интегрирование внешнего источника в STEP7.

В качестве исходного файла выбираем созданный ранее AWL-файл. На следующей картинке приведено окно выбора файла:

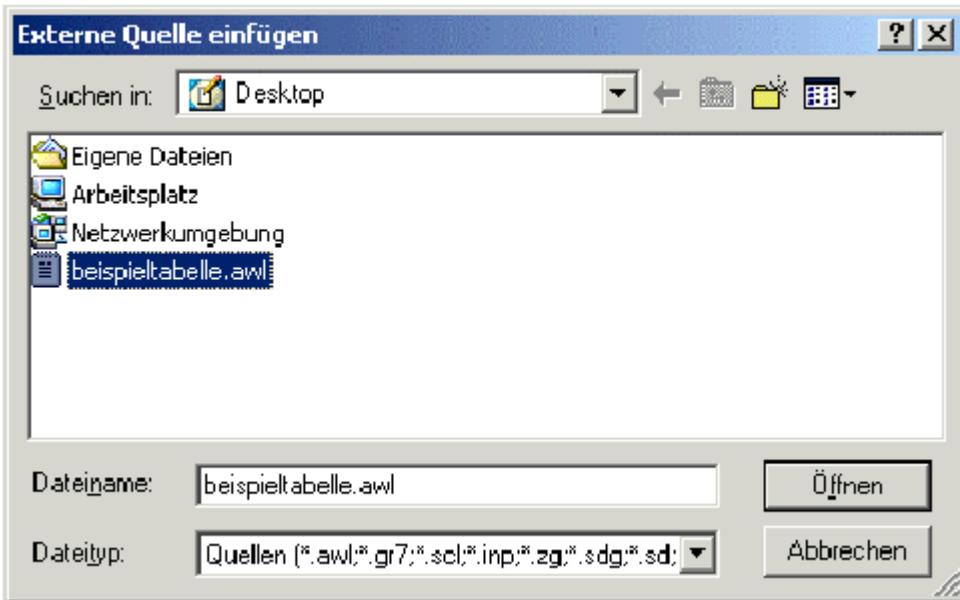


Рис. 3-81. Выбор файла как внешнего источника для интегрирования в STEP7.

Выбранный файл открывается (здесь: "beispieltabelle.awl"). Теперь он существует в качестве исходного файла для проектирования как "источник" (папка "Quellen"). Здесь он вновь выбирается и открывается.

На следующем рисунке показан имеющийся в качестве источника (в папке "Quellen") файл примера, а также связанное с ним контекстное меню:

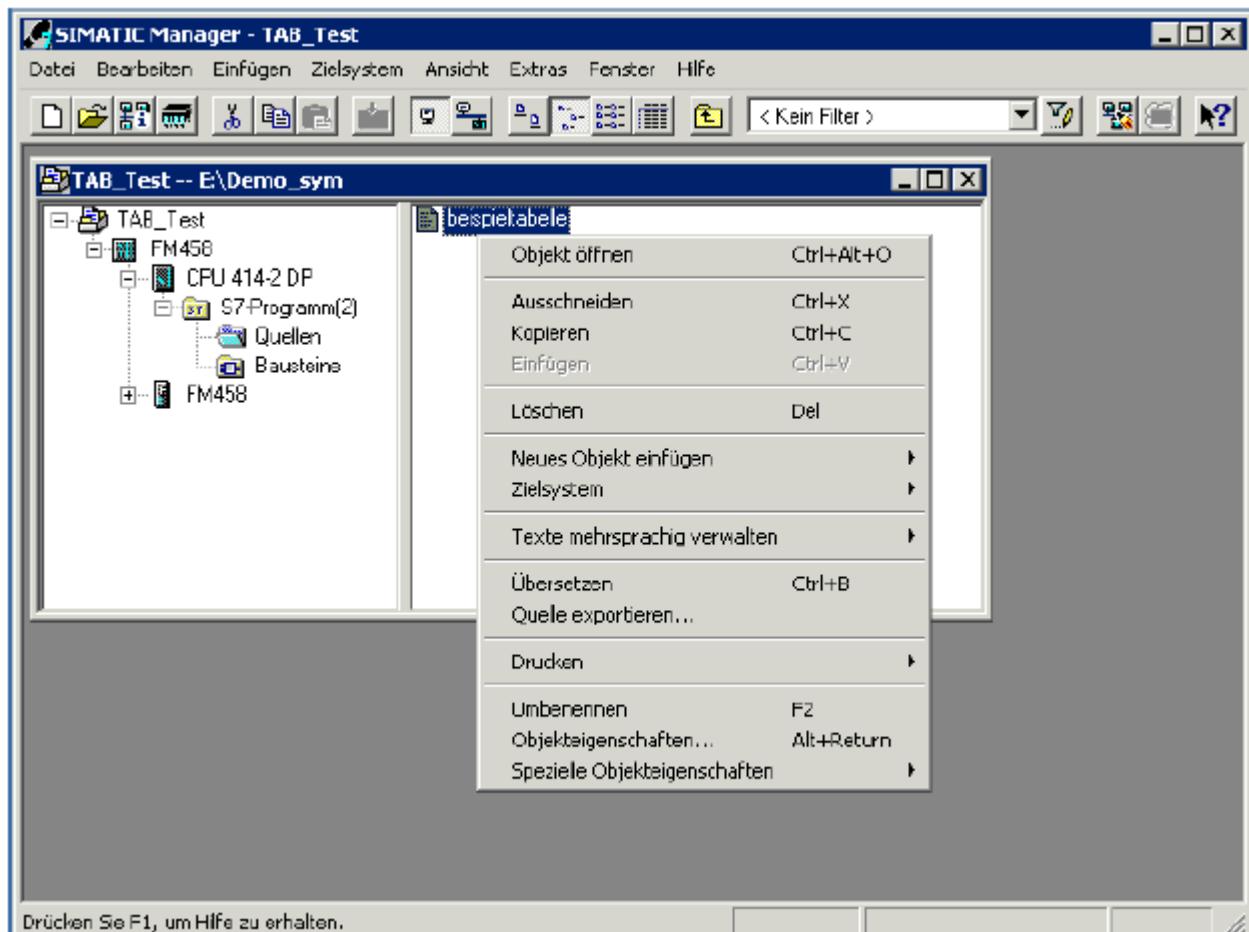


Рис. 3-82. Созданный исходный файл в STEP7.

После открытия файл будет готов к обработке программным пакетом "KOP/FUP/AWL". Здесь возможна лишь компиляция файла посредством "Файл/компилировать" ("Datei/Übersetzen").

Следующий рисунок показывает последовательность действий:

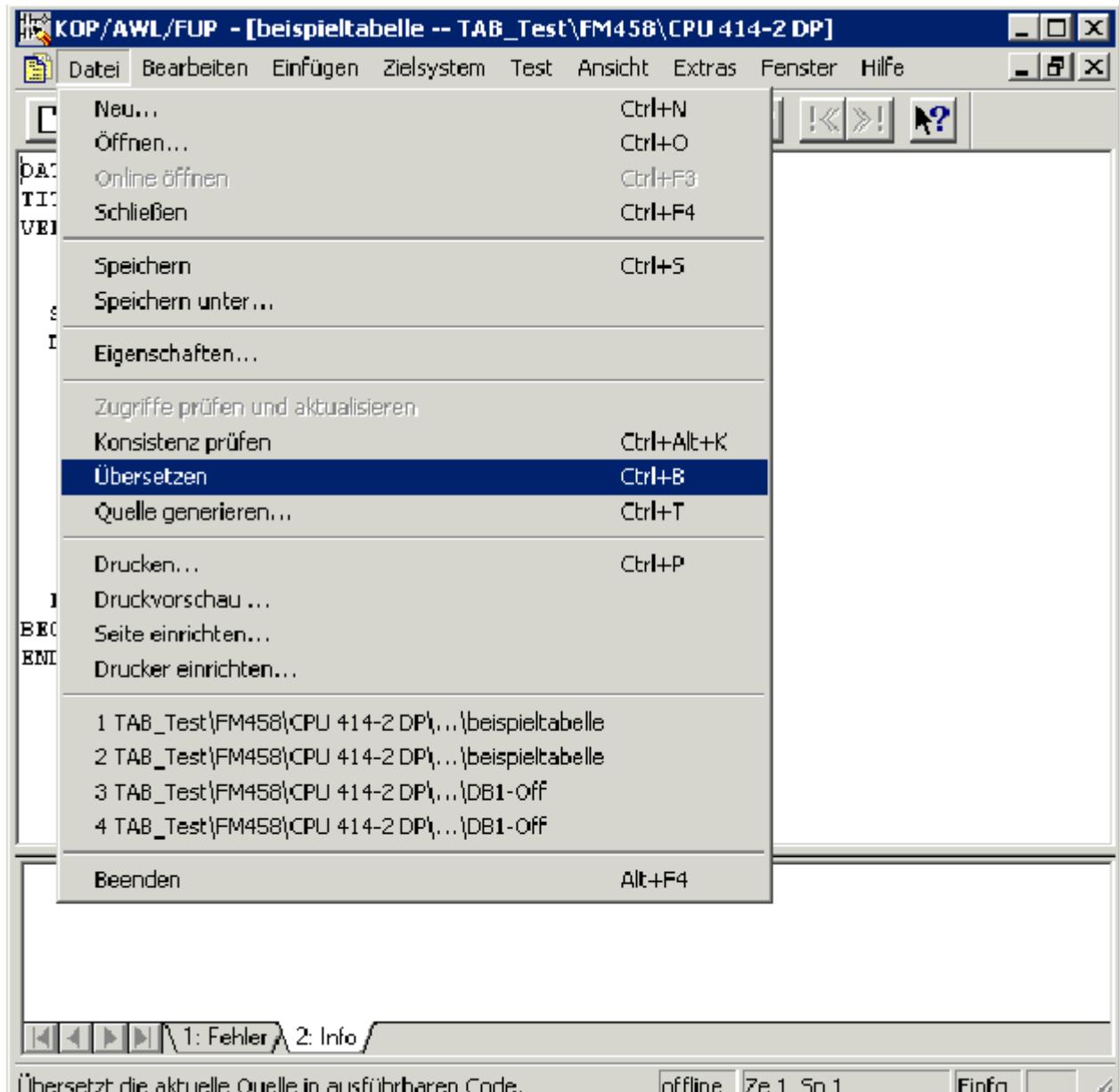


Рис. 3-83. Компиляция исходного файла в приложении "KOP/FUP/AWL".

После успешной компиляции файла в проекте в ваше распоряжение представляется новый DB. Имя нового DB соответствует имени, заданному в начальной строке файла.

Следующая иллюстрация показывает вновь сгенерированный в проекте STEP7, папка "Модули" ("Bausteine"), DB.

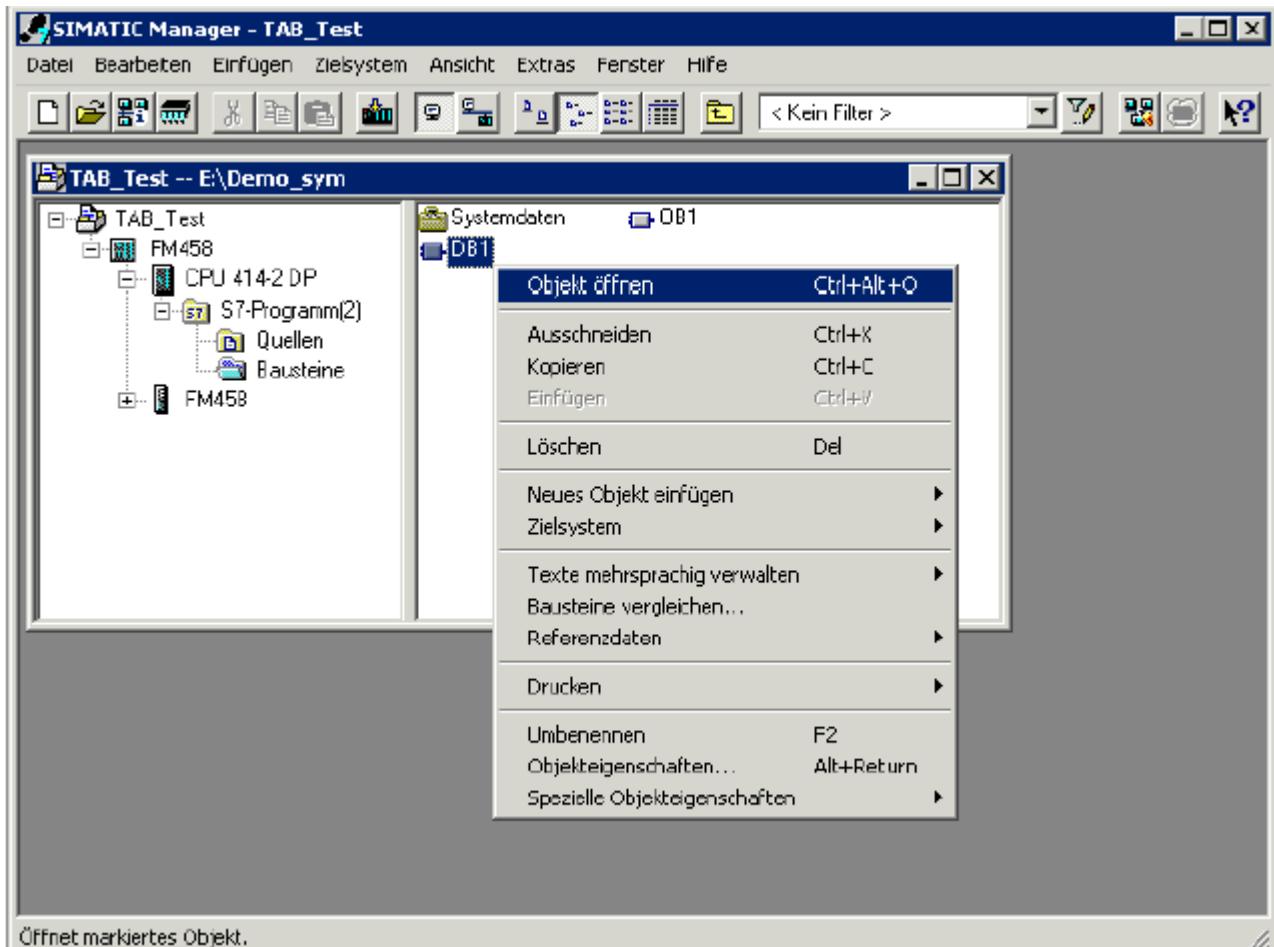
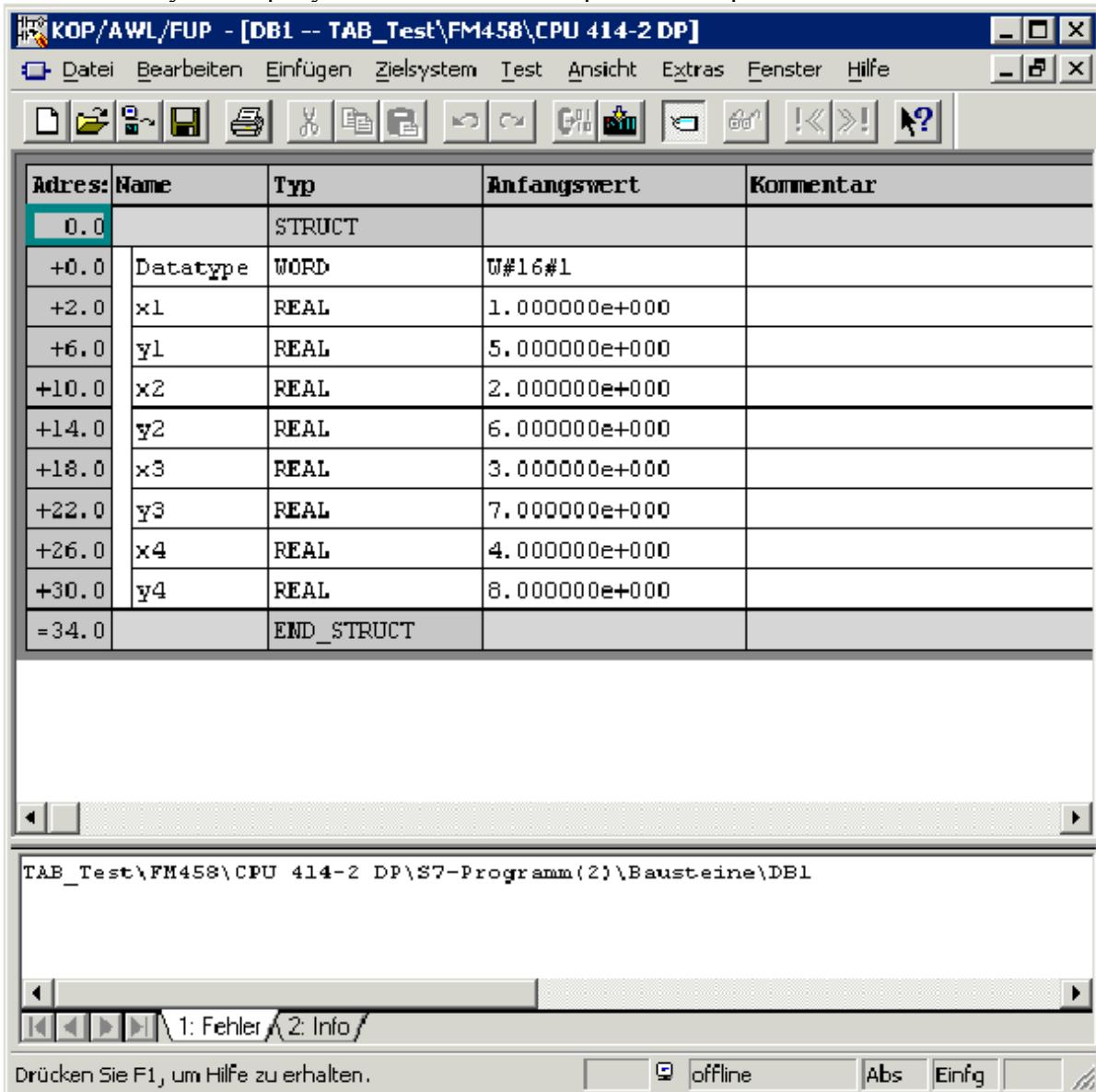


Рис. 3-84. Вновь созданный, после компиляции исходного файла, DB.

Для проверки содержимого DB, его можно открыть в приложении "KOP/FUP/AWL". Для просмотра начальных, а также актуальных значений в меню "Вид" ("Ansicht") выбрать "Вид данных" ("Datenansicht").

Следующий рисунок показывает содержимое открытого DB:



Adress	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	Datatype	WORD	W#16#1	
+2.0	x1	REAL	1.000000e+000	
+6.0	y1	REAL	5.000000e+000	
+10.0	x2	REAL	2.000000e+000	
+14.0	y2	REAL	6.000000e+000	
+18.0	x3	REAL	3.000000e+000	
+22.0	y3	REAL	7.000000e+000	
+26.0	x4	REAL	4.000000e+000	
+30.0	y4	REAL	8.000000e+000	
=34.0		END_STRUCT		

Рис. 3-85. Содержимое вновь созданного DB в приложении "KOP/FUP/AWL".

3.18.3.3.3 Загрузка табличных значений в DB

Если загрузка табличных значений в DB проводится в случае, когда таблица велика, или для нескольких DB недостаточно пользовательской памяти, табличные значения переносятся на прикладной модуль SIMATIC FM 458 несколькими частями. Для этого таблица должна быть разделена на несколько частей. При этом размер каждой части таблицы должен быть выбран таким, чтобы не был превышен объем пользовательской памяти, представленный на S7-CPU. Затем отдельные части таблицы переносятся друг за другом.

ЗАМЕЧАНИЕ Обязательно следует обращать внимание на то, чтобы отдельные части таблицы передавались в возрастающей для пар значений последовательности. Неверная последовательность (передачи) приведет к тому, что табличные значения будут представлены в проекте некорректно.

Имеются две возможности:

- Поочередное заполнение вручную отдельных частей таблицы для DB в приложении "KOP/FUP/AWL" с последующей передачей каждой части таблицы
- Генерация отдельных исходных файлов с различными именами для каждой части таблицы и соответствующее поочередное их встраивание в DB с последующей передачей

Заполнение вручную

Для загрузки табличных значений в DB при ручном заполнении (частей таблицы) необходима следующая последовательность действий:

- Открыть посредством двойного клика мыши соответствующий DB в приложении "KOP/FUP/AWL".
- Имеющиеся табличные значения заменить табличными значениями следующей части таблицы (т. е. части таблицы, подготавливаемой на данном этапе).
- Сохранить DB
- Значения данной части таблицы готовы к передаче

Генерация нескольких исходных файлов

Для загрузки табличных значений в DB при генерации нескольких исходных файлов необходима следующая последовательность действий:

- В заголовке отдельного исходного файла (*.AWL) задается уникальное для каждого файла имя DB.
- Размер отдельного исходного файла не должен превышать объем памяти DB.
- Имена DB желательно нумеровать в возрастающей последовательности.
- Отдельные файлы должны создаваться как исходные только так, как уже описано выше. Они не должны быть скомпилированы.
- Компилируется первый исходный файл, и затем принадлежащие ему табличные значения переносятся в DB.
- Компилируется следующий исходный файл, теперь его табличные значения поставлены в соответствие DB. Теперь они будут перенесены на S7-процессор.
- Аналогичным образом друг за другом компилируются и передаются следующий исходные файлы.
- При передаче последней части таблицы коннектор LASTDB переустанавливается (вручную, пользователем) с 0 на 1. Это является сигналом о завершении передачи.

3.18.3.4 Формирование телеграмм данных при связи по протоколам TCP/IP или DUST1.

Если речь идет о коммуникационном интерфейсе с использованием протоколов TCP/IP или DUST1, следует обращать внимание на формирование телеграммы данных. Это будет описано ниже. Телеграмма данных "создается" функциональными модулями СТВ и СРВ.

Телеграмма данных может быть задана таким образом, что все табличные значения будут перенесены в одном или в нескольких блоках данных.

Следующая таблица показывает формат блока данных:

Тип данных	Описание
char [4]	Идентификатор телеграммы идентифицирует каждую табличную телеграмму маркером "ТАВ0"
u_int16	Команды для телеграммы (битовая кодировка) 1: Новая таблица (нарастающий фронт от 0→1) 2: Конец таблицы
u_int16	Формат данных (REAL=1, DINT=2)
u_int32	№ текущего блока данных
u_int32	Число табличных значений (величин X и Y) Число значений всегда должно быть четным. Т. е. всегда должно переноситься одинаковое число значений X и Y.
u_int32 [56] / float [56]	Область табличных значений. (Значения X и Y всегда чередуются)

Для каждого принятого блока данных ТАВ или ТАВ_D посыпают передатчику подтверждение.

Следующая таблица показывает формат подтверждающей телеграммы:

Тип данных	Описание
char [4]	Идентификатор телеграммы идентифицирует каждую табличную телеграмму маркером "ТАВ0"
u_int32	№ текущего блока данных
u_int32	Статус/номер сбоя 0xB210 OK (блок данных в порядке)

ЗАМЕЧАНИЕ

Новые табличные данные будут перенесены в пассивную таблицу только тогда, когда будет установлена команда "Новая таблица".

После приема команды "Конец таблицы", все другие табличные данные будут отклонены, пока вновь не будет получена команда "Новая таблица".

3.18.4 Автоматическое пользование: карточка памяти

При помощи D7-SYS additionalComponentBuilder (включен в D7-SYS V5.2 плюс SP1) табличные значения могут быть присоединены в качестве дополнительного объекта к компонентам, загружаемым в карточку памяти. Оттуда они могут быть считаны посредством функциональных модулей TAB или TAB_D.

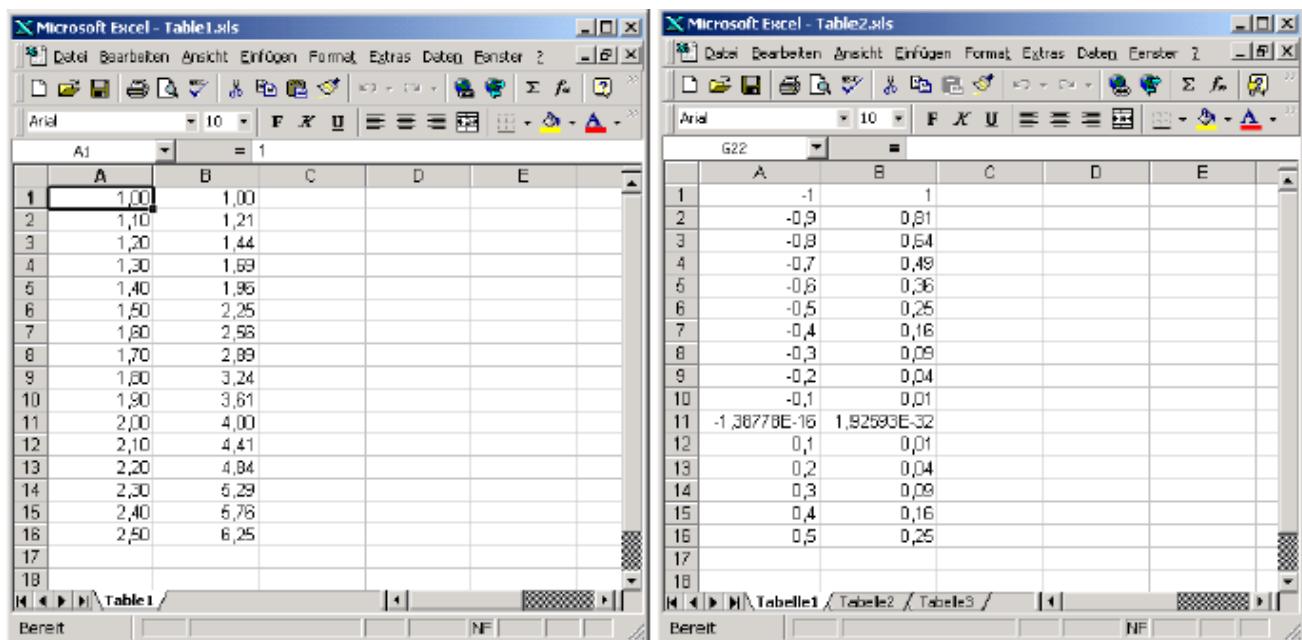
Один или несколько табличных файлов импортируются в D7-SYS additionalComponentBuilder, который присоединяет эти файлы к файлам компонент (загружаемым файлам), которые могут быть загружены в карточку памяти.

D7-SYS additionalComponentBuilder (aCB) принципиально не проверяет содержимое файлов. Таблицы являются исключением из этого правила. Содержание файлов таблиц проверяется. При неверном формате табличного файла aCB выдает соответствующее сообщение.

В следующих разделах на основе примера разъясняется принцип действия от создания файла таблицы до проектирования функционального модуля.

3.18.4.1 Создание табличного файла в формате csv.

Табличные значения создаются любым приложением для создания таблиц (например, MS Excel).



	A	B	C	D	E
1	1,00	1,00			
2	1,10	1,21			
3	1,20	1,44			
4	1,30	1,69			
5	1,40	1,96			
6	1,50	2,25			
7	1,60	2,56			
8	1,70	2,89			
9	1,80	3,24			
10	1,90	3,61			
11	2,00	4,00			
12	2,10	4,41			
13	2,20	4,84			
14	2,30	5,29			
15	2,40	5,76			
16	2,50	6,25			
17					
18					

	A	B	C	D	E
1	-1	1			
2	-0,9	0,81			
3	-0,8	0,64			
4	-0,7	0,49			
5	-0,6	0,36			
6	-0,5	0,25			
7	-0,4	0,16			
8	-0,3	0,09			
9	-0,2	0,04			
10	-0,1	0,01			
11	-1,38778E-16	1,92593E-32			
12	0,1	0,01			
13	0,2	0,04			
14	0,3	0,09			
15	0,4	0,16			
16	0,5	0,25			
17					
18					

Рис. 3-86. Табличные значения в Excel.

Условия

Файлы таблицы должны удовлетворять следующим условиям:

- Табличный файл принципиально должен содержать два столбца; если таблица будет содержать большее число столбцов, в диалоговом окне будет выдано сообщение об ошибке.
- Оба столбца должны содержать одинаковое число значений. В противном случае D7-SYS additionalComponentBuilder выдаст в диалоговом окне сообщение об ошибке и табличные значения не будут приняты в обработку.

D7-SYS additionalComponentBuilder ожидает следующие форматы данных:

При описании типа¹ "Таблица REAL" ("Table REAL"):

- [+/-]xxx.yyy – значения Real, десятичная запятая задается посредством "." (например 145.123)
- [+/-]xxx.yyy – значения Real, десятичная запятая задается посредством "," (например 145,123)
- [+/-]xxx.yyyE+/-mm – значения Real в экспоненциальном представлении, десятичная запятая задается посредством "." (например 145.122 E+12)
- [+/-]xxx.yyyE+/-mm – значения Real в экспоненциальном представлении, десятичная запятая задается посредством "," (например 145,122 E+12)

При описании типа¹ "Таблица DINT" ("Table DINT"):

- [+/-]xxx – Integer или Double-Integer (например 145)

Кроме того, табличные файлы должны удовлетворять следующим условиям:

- ASCII – файлы
- Разделение колонок таблицы посредством точки с запятой или табуляции
- Разделение строк посредством знака "перевод строки" (Enter) либо точки с запятой

¹ Имеется ввиду окно **Typbeschreibung** в диалоговом окне **Eigenschaften** – свойства компонента (см. стр. 33)

Сохранение таблиц

Этим условиям удовлетворяют таблицы, созданные в MS Excel, и сохраненные в формате CSV (разделители – запятые) (*.csv) или в формате "Текстовые файлы (с разделителями табуляции) (*.txt)".

Следующая иллюстрация показывает примеры двух файлов с табличными значениями, которые были сохранены в csv-формате:

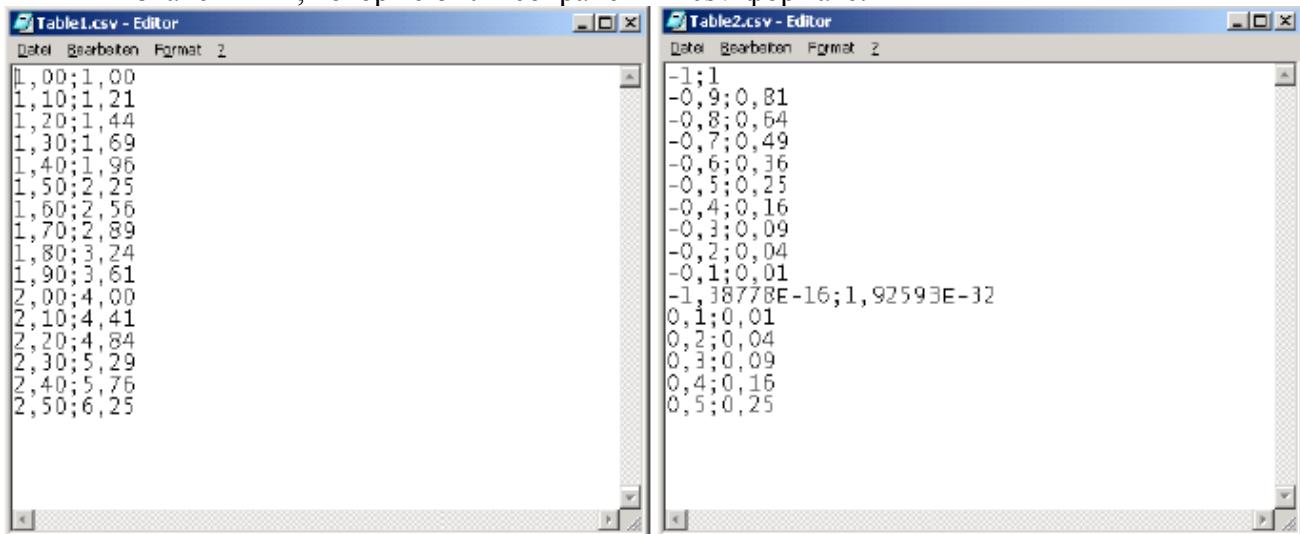


Table1.csv - Editor

Table2.csv - Editor

Table1.csv	Table2.csv
1,00;1,00	-1;1
1,10;1,21	-0,9;0,81
1,20;1,44	-0,8;0,64
1,30;1,69	-0,7;0,49
1,40;1,96	-0,6;0,36
1,50;2,25	-0,5;0,25
1,60;2,56	-0,4;0,16
1,70;2,89	-0,3;0,09
1,80;3,24	-0,2;0,04
1,90;3,61	-0,1;0,01
2,00;4,00	-1,38778E-16;1,92593E-32
2,10;4,41	0,1;0,01
2,20;4,84	0,2;0,04
2,30;5,29	0,3;0,09
2,40;5,75	0,4;0,16
2,50;6,25	0,5;0,25

Рис. 3-87. Табличные значения, разделенные точкой с запятой (формат csv).

3.18.4.2 Работа с D7-SYS additionalComponentBuilder.

После того, как файлы таблиц были сохранены в формате csv, они могут быть импортированы в D7-SYS additionalComponentBuilder.

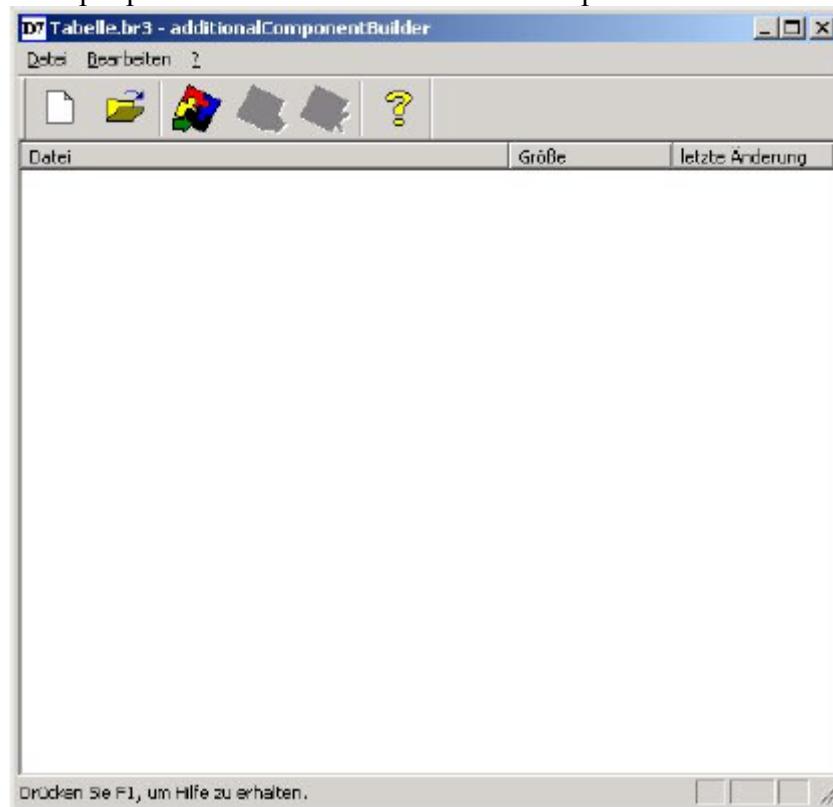


Рис. 3-88. D7-SYS additionalComponentBuilder.

На последнем этапе посредством  закладывается новый файл компонент.

Новый компонент



Рис. 3-89. Определение свойств (*Eigenschaften*).

Должны быть проведены следующие определения (установки):

Эти установки не могут быть изменены позднее и после их проведения будут представляться на сером фоне.

- **Версия D7-SYS (D7-SYS Version)**

Текстовое окошко, в котором задается версия, для которой будет создан компонент

- **Тип компонента (Komponententyp)**

Текстовое окошко с предопределенными элементами "USER", "IT1" и "IT2". Значение по умолчанию – "USER".

Элементы имеют следующее значение:

- USER – создаваемый пользователем файл компонент, например файл таблицы
- IT1/IT2 – системный файл компонент для "железного" модуля ITSP (Simadyn D).

- **Описание типа (Typbeschreibung)**

Текстовое окошко для задания "Table REAL" или "Table DINT". Значение по умолчанию для типа компонента "USER" – "Table REAL". "Table DINT" применяется для таблиц формата DINT.

Элементы имеют следующее значение:

- Table REAL – файл таблицы, содержащий данные типа REAL
- Table DINT – файл таблицы, содержащий данные типа DINT

(Новое описание типа вносится в текстовое окошко и подтверждается Enter. Это новое описание типа воспринимается в текстовом окошке и в последующем может в нем выбираться)¹.

Сохранение

С завершением задания свойств завершается создание нового файла компонент. Новый файл компонент стандартно (по умолчанию) размещается в директории C:\temp. Если при запоминании будет определен путь к другому каталогу, при новом старте программы уже путь к этому каталогу будет принят по умолчанию.

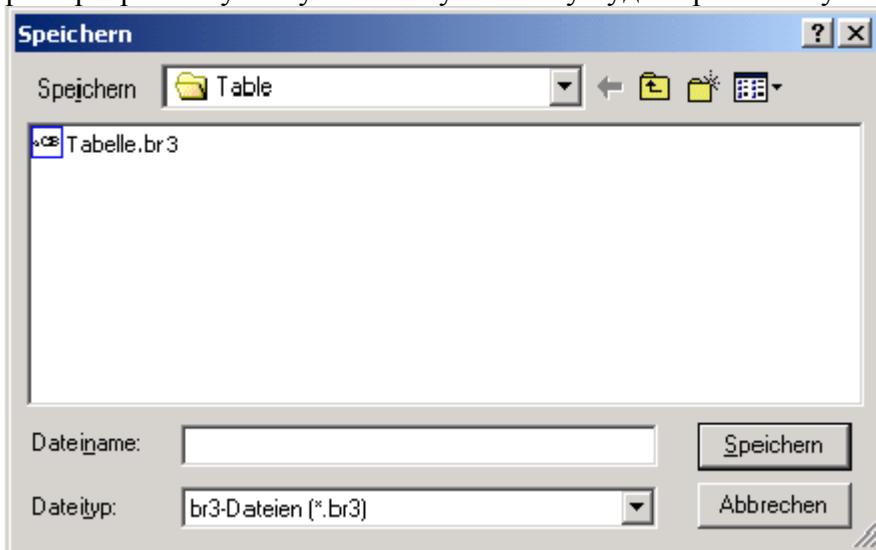


Рис. 3-90. Запоминание нового файла компонент.

Теперь можно добавлять файлы таблиц. Посредством  открывается окно выбора файлов, где можно отметить желаемый табличный файл.

ЗАМЕЧАНИЕ Компонент с описанием типа "Table" может содержать только таблицу, значения которой имеют соответствующий формат! То есть таблица REAL (Table REAL) содержит только таблицу со значениями типа REAL.

¹ Заключенный в скобки дословный перевод требует некоторых пояснений – изложено довольно кратко.

После выбора свойств нового компонента из предлагаемых вариантов и подтверждения выбора (кнопкой OK) будет предложено сохранить файл компонент. Тип файла задан: "*br3-Dateien (*.br3), имя файла определяет пользователь (например, Tabelle3). Имя файла компонент лучше задавать отличным от имени файла таблицы, так как один файл компонент может содержать несколько табличных файлов. После задания имени файла и нажатия

кнопочки "Сохранить" в additionalComponentBuilder активизируется кнопочка  (добавить). До тех пор, пока файл таблицы (*.csv) не будет добавлен в новый файл компонент (*.br3) Вы не обнаружите файл компонент нигде (а ведь он был сохранен!). "Пустой" файл компонент не будет создан – daß ist unmöglich (не MS Word какой-то!). Поэтому приведенную на рис. 3-90 картинку Вы увидите лишь после добавления файла таблицы в файл компонент.

Следующая иллюстрация показывает содержимое D7-SYS additionalComponentBuilder после импорта обеих созданных в качестве примера табличных файлов:

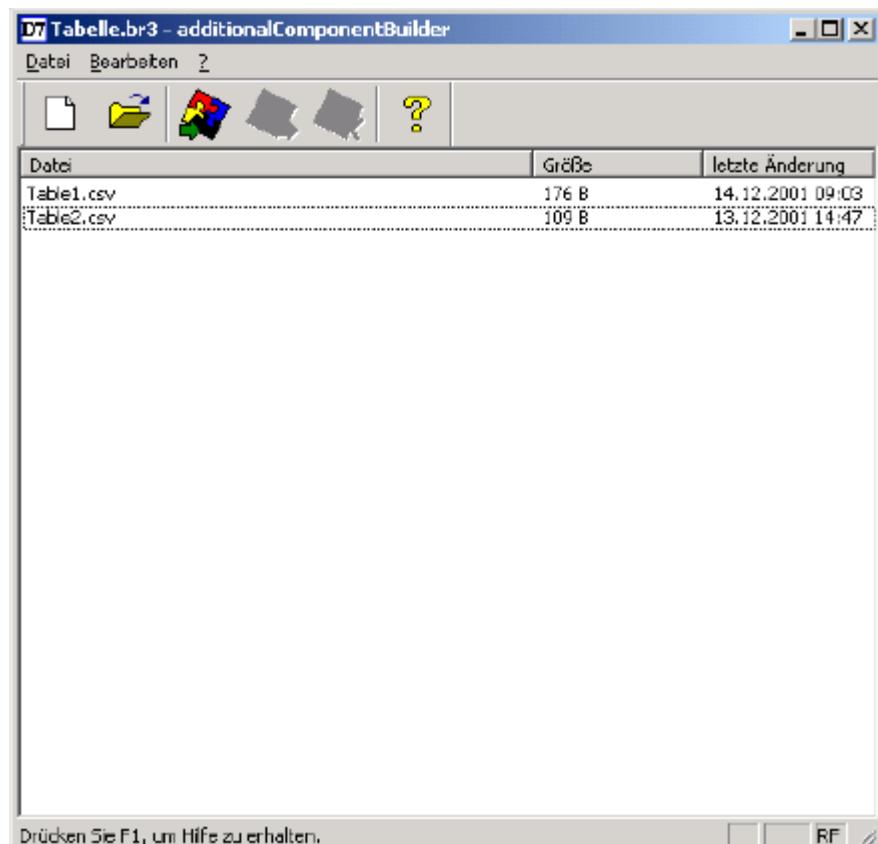


Рис. 3-91. D7-SYS additionalComponentBuilder с импортированными табличными файлами.

В любое время другие табличные файлы могут добавляться, импортироваться или удаляться. D7-SYS additionalComponentBuilder автоматически воспринимает выбор табличного файла и запоминает модифицированный файл компонент.

Открывание

При открывании существующих компонент D7-SYS additionalComponentBuilder по умолчанию обращается к папке "C:\temp". Если будет выбран путь к другому каталогу, при новом старте программы этот путь будет воспринят по умолчанию.

3.18.4.3 Загрузка

После создания посредством D7-SYS additionalComponentBuilder файла компонент в общем случае можно открывать диалог загрузки.

(1) Открытие диалога загрузки в D7-SYS посредством "Система назначения → Загрузить" ("Zielsystem → Laden")

Посредством этого диалога можно загрузить (Off-/Online) опциональные компоненты текущего проекта в карточку памяти.

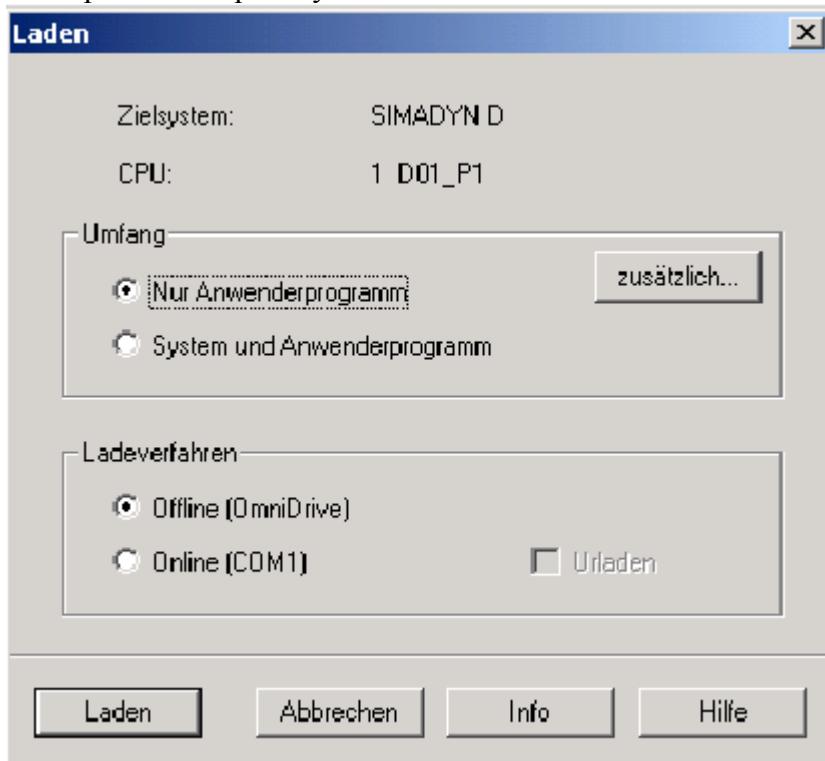


Рис. 3-92. Диалог загрузки для "Zielsystem → Laden" в D7-SYS.

(2) Открывание диалога для опциональных компонент.

Могут быть выбраны максимум два компонента. Кликом по кнопке "Новый" ("NEU...") выбирается файл для выбранного компонента.



Рис. 3-93. Диалог выбора опциональных компонент, например, табличного файла.

(3) Открывается диалог выбора файла для выбранного дополнительного компонента.

Для созданного заранее посредством D7-SYS additionalComponentBuilder файла компонент может быть выделен и записан в процессе загрузки в карточку памяти только компонент IT1.

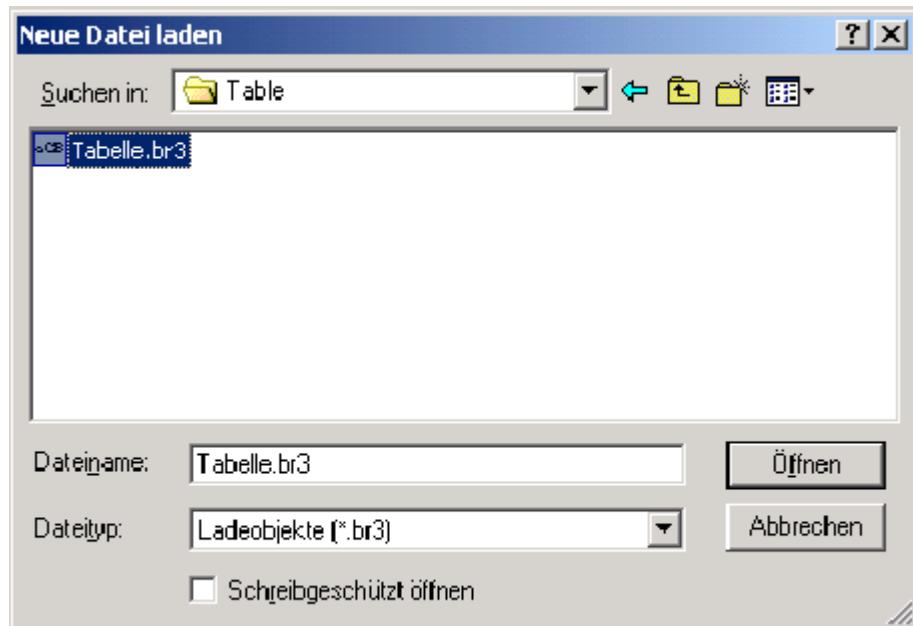


Рис. 3-94. Загрузка файла компонент.

3.18.4.4 Проектирование функциональных модулей

Для способа пользования "Автоматическое пользование: карточка памяти" должны проектироваться только функциональные модули TAB или TAB_D, в зависимости от того, табличными значениями какого типа данных, REAL и/или DIINT, они будут управлять. Каждая таблица должна содержать данные одного типа. Если нескольким таблицам поставлены в соответствие данные различного типа, TAB или TAB_D должен быть спроектирован для каждой таблицы.

Функциональные модули TAB или TAB_D должны быть спроектированы в задаче со временем опроса больше или равным 32mS. Необходимы следующие объявления на коннекторах:

CTS = 0

US = не объявляется

NAM = имя файла таблицы (с расширением имени файла, заданном при сохранении, например в MS Excel)

AUT = 1 (активирована автоматическая работа)

Следующий рисунок показывает проектирование:

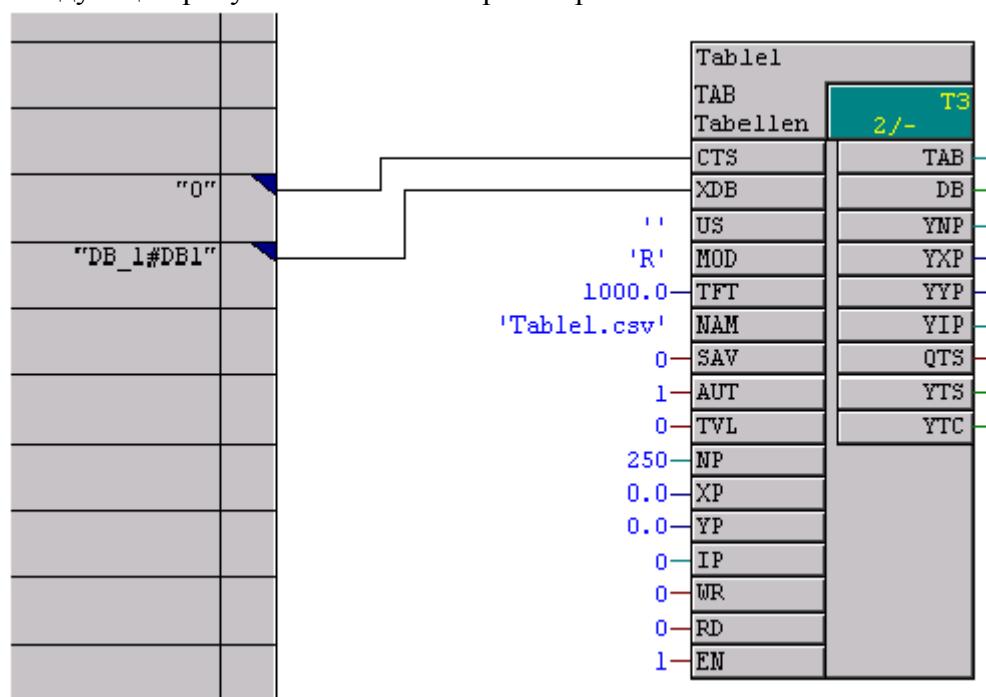


Рис. 3-95. Проектирование функционального модуля TAB.

На следующем рисунке представлены табличные функциональные модули для двух таблиц. Табличные величины, подчиненные функциональным модулям, могут быть использованы только следующими (другими) функциональными модулями, например, FB TABSAM.

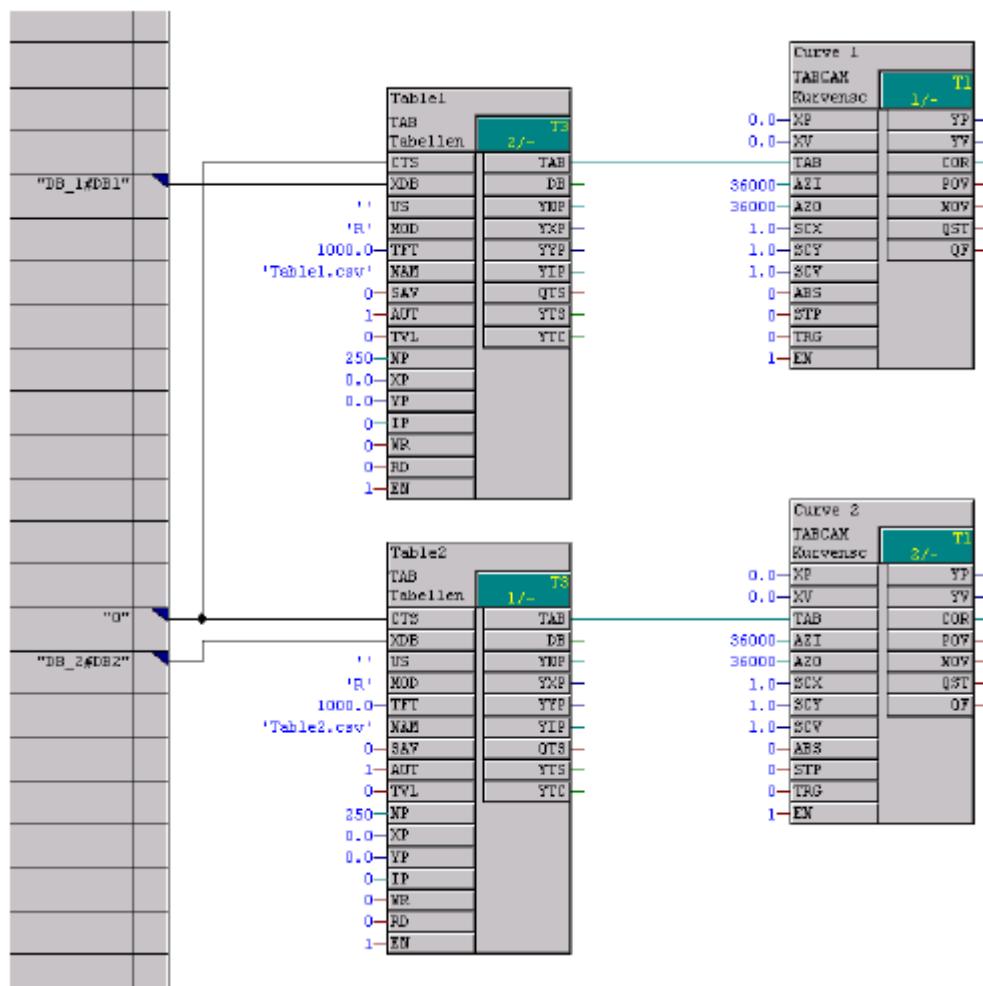


Рис. 3-96. Пример проектирования.

3.19 Техника доступа к параметрам в D7-SYS

3.19.1 Общее описание функциональности параметров

Общее

Параметрирование позволяет Вам при помощи устройства для обслуживания производить следующие операции над параметрами на коннекторах программных модулей:

- Читать значения
- Изменять значения
- Изменять значения с запоминанием изменений в памяти CPU
- Изменять связи при помощи ВІСО-техники
- Изменять связи с запоминанием изменений в памяти CPU
- Читать элементы описания параметров

Аппаратное обеспечение

Вы можете использовать технику доступа к параметрам для следующего аппаратного обеспечения:

- Технологический модуль T400
- Прикладной модуль FM 458
- Стандартные CPU Simadyn D

ЗАМЕЧАНИЕ

Под поддерживающими параметрирование служебными устройствами понимаются служебные устройства Masterdrive, например OP1S или DRIVE ES/DRIVE Monitor.

3.19.1.1 Параметр

Для реализации техники доступа к параметрам в D7-SYS Вы обозначаете входа или выхода программных модулей как параметры.

Различают два типа параметров:

- **Наблюдаемый параметр**
 - Может быть спроектирован для входов или выходов программных модулей
 - Значения могут только считываться
- **Устанавливаемый параметр**
 - Может быть спроектирован на входах программных модулей
 - Значения могут считываться, изменяться и запоминаться в изменяемой памяти
 - Посредством BICO-техники могут изменяться связи с другими программными модулями

ЗАМЕЧАНИЕ

Если на входах программных модулей спроектированы \$-переменные или виртуальные связи, Вы не сможете изменять значения параметров.

Типы данных коннекторов¹ для параметров

В D7-SYS могут быть спроектированы в качестве параметров коннектора программных модулей со следующими типами данных:

Тип данных коннектора в CFC	Bool	Integer	Double Integer	Word	Real	SDTime
Тип параметра в его описании	Q2	I2	I4	V2	I4	I4

Проектирование параметров

В распоряжении пользователя имеется до 2000 различных параметров. Каждый параметр должен быть задан только один раз. Параметр должен быть спроектирован в CFC следующим образом:

Обозначьте коннектор программного модуля псевдокомментарием @TP_bnnn, где:

- b: Идентификатор области "H", "L", "c" или "d"
 - обозначает область номера параметра
 - "H" или "L": параметр может считываться и изменяться
 - "c" или "d": параметр может только считываться
- nnn: трехзначный номер параметра
 - от 000 до 999

¹ Каждый библиотечный программный модуль (подпрограмма) имеет заранее определенный список входных и выходных переменных. Каждой переменной отведена область памяти (коннектор). Каждой переменной определен формат (тип данных).

ЗАМЕЧАНИЯ

- Конкретный номер параметра должен быть определен только один раз (проверка в CFC)
- Псевдокомментарий не может быть спроектирован для интерфейсного коннектора плана¹
- На коннекторе программного модуля плана, который должен быть скомпилирован "как модуль"¹, не может быть задан псевдокомментарий
- На одном коннекторе программного модуля не может быть определено посредством псевдокомментария более одного параметра
- Комментарий может содержать несколько псевдокомментариев, отделенных друг от друга, а также от "нормального" текста, пробелами; например: "@TP_H098 @DATX это технологический параметр"

Доступ к параметрам

Вы можете получить доступ к параметрам извне (например, *(к параметрам)* от вышестоящей системы регулирования Simadyn D) следующим образом:

Примечание переводчика. Пояснение довольно кратко. Важно понять: объявляя параметры в вышестоящей системе Simadyn D (или FM 458 в составе Simatic S7-400, или T400 в составе SRT400) мы открываем возможность, подключившись к ведомому приводу пакетом Drive Monitor или непосредственно с OPIS, "достучаться" до конкретного параметра этой вышестоящей системы и даже изменить связь в CFC-проекте!!!

Псевдокоммент.	Техноплата T400 Показания служебного устройства	T400 базовое устройство/модуль CPU в корзине Simadyn D Показания служебного устройства	Возможно проектирование на коннекторе Е: вход А: выход	Изменение связи	Значение
@TP_dxyz	dxyz	gxyz	A/E	любое	Наблюдаемый параметр
@TP_cxyz	cxyz	nxyz	A/E	любое	Наблюдаемый параметр
@TP_Hxyz	Hxyz	Rxyz	E	Нет или OP-связь	Устанавливаемый параметр
@TP_Lxyz	Lxyz	Uxyz	E	Нет или OP-связь	Устанавливаемый параметр
@TP_Hxyz	Hxyz	Rxyz	A	любое	Наблюдаемый параметр
@TP_Lxyz	Lxyz	Uxyz	A	любое	Наблюдаемый параметр

Легенда

xyz: номер параметра

любое: связь может изменяться или не изменяться

OP-связь: переключение между глобальными операндами

¹ Речь идет о создании CFC-макроса

3.19.1.2 BICO-техника

При помощи BICO-техники Вы можете изменять связи между программными модулями со служебной панели Masterdrives. Тем самым открывается возможность изменения проекта без CFC. Вы сможете изменять связи (между программными модулями CFC-планов) техноплаты T400, прикладного модуля FM458 или модуля CPU в корзине SIMADYN D.

ОПАСНОСТЬ

- Не могут работать одновременно функции BICO-техники и тестовый режим CFC.
- Если в тестовом режиме CFC были проведены online-изменения, необходимо вначале скомпилировать программу, и лишь затем перейти к использованию функций BICO-техники. Изменения CFC будут доступны для отображения на служебном устройстве только после компиляции.
- Если средствами BICO-техники были предприняты изменения без запоминания в памяти изменений CPU, не гарантируется согласованность данных между изменениями в CPU и Вашим проектом; согласование данных не будет также проведено и при актуализации проекта. Для того чтобы избежать состояния несогласованности данных, необходимо, перед переводом CFC в тестовый режим, провести RESET (аппаратный сброс) модуля CPU.

ЗАМЕЧАНИЕ

Если Вы вначале провели изменение связей посредством BICO-техники, а затем активировали тестовый режим CFC, в качестве предупреждения появляется диалог "Измененное состояние программного обеспечения".

Типы данных для технологических коннекторов

В D7-SYS могут быть спроектированы в качестве технологических коннекторов коннекторы со следующими типами данных:

Тип данных коннектора в CFC	Bool	Integer	Double Integer	Word	Real	SDTime
Тип данных технологического коннектора в его описании	O2	I2	I4	V2	I4	I4

Проектирование технологических коннекторов

Для того чтобы стало возможным изменение связей между программными модулями посредством BICO-техники, проектирование параметров на выходах программных модулей в CFC необходимо дополнить проектированием технологических коннекторов. Выходы программных модулей, объявленные технологическими коннекторами, Вы можете использовать для изменения связей средствами BICO-техники.

Технологический коннектор проектируется (объявляется) следующим образом:

Обозначьте выход программного модуля псевдокомментарием @TC_nnnn, где

- nnnn – четырехзначный номер технологического коннектора
– от 0000 до 999
- Для конкретного выхода программного модуля можно, посредством псевдокомментария, спроектировать не более одного технологического коннектора.
- Конкретный номер технологического коннектора может быть определен только один раз (проверяется CFC).
- Технологический коннектор не может быть спроектирован на интерфейсном коннекторе плана
- Технологический коннектор не может быть спроектирован на коннекторе программного модуля CFC-плана, который (план) будет компилироваться "как модуль"
- Комментарий может содержать несколько псевдокомментариев, отделенных друг от друга, а также от "нормального" текста, пробелами; например: "@TC_1398 @TP_H345 технологический параметр 345 объявлен технологическим коннектором 1398"

ЗАМЕЧАНИЯ

Чтение параметров

На служебной панели¹ Вы можете считывать параметр и его значение.

Считанное значение соответствует:

- Для коннектора программного модуля, связанного с технологическим коннектором, – номер технологического коннектора @TC_nnnn.
- Для коннектора программного модуля, не связанного с технологическим коннектором, – значение переменной на входе или выходе.

Из документации на параметры стандартного проекта² вы узнаете, что представляет собой считанное значение: номер технологического коннектора или собственно величину параметра. Отметить эти различия при индикации на служебной панели не представляется возможным.

Изменение связей посредством BICO-техники

Посредством BICO-техники Вы можете изменять существующие связи между программными модулями только в том случае, если эти связи были спроектированы в CFC следующим образом:

- На выходах программных модулей посредством псевдокомментариев @TC_nnnn были спроектированы технологические коннекторы.
- На входах программных модулей посредством псевдокомментариев @TP_Hnnn или @TP_Lnnn были спроектированы параметры.

¹ Напомним, что под служебной панелью в данном случае имеется ввиду панель OP1S или PC с установленным пакетом Drive Monitor. Например, мы "рассматриваем" с OP1S технологический параметр, объявленный в CFC-проекте техноплаты T400, установленной в электронный бокс Simoreg 6RA70.

² Имеются ввиду продаваемые Siemens готовые (стандартные) проекты для T400 в составе привода, например, моталки ("Simoreg DC Master. Applikation Achswickler"), Parameterliste которого ("achswickler.dnl") Вы сможете обнаружить на CD-ROM Simoreg DC Master (зак. № 6RX1700-0AD64) в папке "Applikationen_d".

- Связь между программными модулями была создана как связь между выходом с псевдокомментарием `@TC_nnnn` и входом с псевдокомментарием `@TP_Hnnn` или `@TP_Lnnn`.

BICO-техника поменяет связь, когда Вы со служебной панели зададите в качестве значения параметра другой номер технологического коннектора `@TC_nnnn`.

- Максимальное число возможных посредством BICO-техники изменений связей на различных входах, которые будут запомнены в памяти изменений, составляет:
 - для технологического модуля T400: примерно 1600
 - для прикладного модуля FM458: примерно 400
 - для модуля CPU в корзине Simadyn D: примерно 400
- При помощи BICO-техники можно изменять, но не удалять, только уже имеющиеся связи между программными модулями.
- При помощи BICO-техники нельзя создавать новые связи для входов, связи к которым не были запроектированы ранее.
- Изменения связей между коннекторами программных модулей, проведенные посредством BICO-техники будут действительны при актуализации CFC-проекта только тогда, когда эти изменения были запомнены¹.
- При изменениях связей между коннекторами программных модулей средствами BICO-техники действительны те же соглашения для проверки типов коннекторов, что и в CFC.

ЗАМЕЧАНИЯ

Псевдокомментарий `@DATX` не поддерживается в тестовом режиме CFC. При изменении связи с входом программного модуля, снабженного псевдокомментарием `@DATX`, значение переменной для этого входа вновь будет актуализироваться в соответствии с механизмом согласования (*т. е. в соответствии со временем опроса – *Abtastzeit* – данного программного модуля*). Псевдокомментарий `@DATX` утрачивает силу.



ОСТОРОЖНО

Подсказка: вновь скомпилируйте и загрузите пользовательскую программу.

¹ Например, запоминание изменений параметров, произведенных с панели OP1S, для случая использования T400 в составе электронного бокса Simoreg 6RA70, инициируется установкой слова управления памятью параметров (P053.001) в "1".

Пример

Возможность переключений в BICO-технике и её назначение

Псевдо-комментарий	Тип коннектора	Связан с	Операции на служебной панели	
			Чтение	Запись
@TP_L/H	Вход	Норма	Отображается значение переменной	Невозможна
@TP_L/H	Вход	Меркер	Отображается значение переменной	Изменение значение переменной
@TP_L/H	Вход	\$-сигнал	Отображается значение переменной	Невозможна
@TP_L/H	Вход	Виртуальная связь	Отображается значение переменной	Невозможна
@TP_L/H	Вход	нет связи	Отображается значение переменной	Изменение значение переменной
@TP_L/H	Вход	@TC_	Отображается номер технол. коннектора @TC_	Задание нового номера @TC_, с которым желаете установить связь
@TP_c/d	Вход	Любая	Отображается значение переменной	Невозможна
@TP_c/d	Вход	@TC_	Отображается значение переменной	Невозможна
@TP_L/H	Выход	Любая	Отображается значение переменной	Невозможна
@TP_c/d	Выход	Любая	Отображается значение переменной	Невозможна
@TC_	Вход	нет связи	Сообщение об ошибке при компиляции в CFC	
@TC	Выход	нет связи	Источник связи в BICO-технике	

Легенда:

@TP_L/H: Параметр @TP_Lnnn или @TP_Hnnn
@TP_c/d: Параметр @TP_cnnn или @TP_dnnn
@TC_: Технологический коннектор @TC_nnnn
Норма: Выход, не являющийся меркером, \$-сигналом или виртуальной связью
Любая: Может быть связан или не связан

Переключения связей между программными модулями различных задач

Число вновь устанавливаемых связей между программными модулями различных задач в BICO-технике ограничено. Для пользователя имеет значение наибольшая из следующих величин:

- Число 20
- 20% от уже спроектированного числа связей между программными модулями различных задач
- 0,25 x (число спроектированных в задаче n технологических коннекторов @TC_....)

3.19.1.3 Зависимые от состояния изменения параметров

Для того чтобы определенный параметр изменялся в функции состояния системы, должны быть спроектированы следующие функциональные программные модули:

- Функциональный модуль PSTAT
 - для проектирования состояния устройства
 - заданием пароля разблокируется уровень доступа
- Функциональный модуль PLIM
 - Для определения состояний и уровней доступа, при которых параметр должен изменяться

Дальнейшая информация

По функциональным модулям смотри руководство "Управляющая система Simadyn D, Библиотека функциональных модулей" ("Regelsystem SIMADYN D, Funktionsbaustein-Bibliothek").

3.19.1.4 Идентификация Simadyn D - компонентов

Зарезервированные параметры

При проведении идентификации Drive Monitor анализирует (использует) параметры d998 (1998) и d999 (1999).

d998	Устройство	Особенности
80	Simadyn D в целом	Речь идет о специализированной для Simadyn D расширенной области параметров для до 16 x 2000 параметров.
134	T400	Возможны параметры в области параметров Основного Устройства (0...999). Номер и значение параметра могут случайным образом совпасть с номером и значением идентификационного параметра микропроцессорного тиристорного преобразователя; в этом случае, идентификатор может указать на несуществующее устройство ¹ .
134	FM458/SRT400	Область параметров = технологические параметры (1000...1999; 3000...3999) Область параметров является функцией установок на центральном FB ² <ul style="list-style-type: none"> • BASEBOARD: 0 ... 999; 2000 ... 2999 • TECHBOARD: 1000 ... 1999; 3000 ... 3999

Последовательность действий при идентификации

Предпосылка: пользователь выбирает Simadyn D или SRT400 и устанавливает online связь с устройством.

В соответствии с выбранным типом устройства Drive Monitor проверяет идентификационный параметр d998. При успешном завершении идентификации цикл повторной проверки с целью распознавания иного устройства не возобновляется.

1. Пользователь выбрал Simadyn D: в случае, когда d998=80, идентификация считается успешно завершенной.
2. Пользователь выбрал SRT400: в случае, когда d998=134, идентификация считается успешно завершенной. Таким образом, пользователь может обращаться только к технологии (модулю/плате технологии), независимо от Основного Устройства!

Далее имеет значение: Параметр d999 – опция для распознавания варианта и версии программного обеспечения стандартного проекта.

d999	Программное обеспечение	Пример
1AB	Синхронное угловое перемещение Version A.Bx (x служит для нумерации соответствующей версии)	120 ♦ SPA440 V2.0x 123 ♦ SPA440 V2.3x
2AB	Моталка Version A.Bx	221 ♦ SPW420 V2.1x
3AB	Поперечный раскрой/управление ножницами Version A.Bx	310 ♦ SPS450 V1.0x

Если идентификация устройства не была успешной, попытайтесь идентифицировать распознанный тип устройства.

Если в Drive Monitor была распознана другое (дословно – отклоняющееся) программное обеспечение, в диалоге "Geraeteidentifikation" ("Идентификация устройства"), будет предложена опция "Datenbasis erzeugen" ("Создать базу данных"). Будет образована специфическая база данных.

¹ По-немецки: "D.h. eine Identifikation kann zu Zufallsprodukten fuehren, wenn ein Parameter zufaellig mit dem Identifikationsparameter und -wert eines Stromrichters uebereinstimmt".

² Центральный FB-установочный коннектор – это: @DRIVE.BBF – для T400 в составе SRT400; @FMPAR.BBF – для FM458.

3.19.1.5 Размерность и обозначение размерностей

Для того чтобы Вы могли снабдить входную или выходную переменную программного модуля размерностью (физическая величина), Вы должны спроектировать на коннекторе (Eigenschaften – Anschluss-Einheit) программного модуля текстовую строку в соответствии с приведенной ниже таблицей.

Физическая величина	Размерность	Проектируемая текстовая строка
Длина	Метр Миллиметр Километр Микрометр	m mm km um
Площадь	Квадратный метр Квадратный миллиметр Квадратный километр	m ² mm ² km ²
Объем	Кубометр Литр	m ³ l
Время	Секунда Минута Час День Миллисекунда Микросекунда	s min h d ms us
Сила	Ньютон Кило Ньютон Мега Ньютон	N kN MN
Давление	Паскаль Кило Паскаль Миллибар Бар	Pa kPa mbar bar
Масса	Килограмм Грамм Миллиграмм Тонна	kg g mg t
Энергия, работа	Джоуль Килоджоуль Мега джоуль Ватт/час Киловатт/час Мегаватт/час	J kJ MJ Wh kWh MWh
Активная мощность	Ватт Киловатт Мегаватт Милливатт	W kW MW mW

Физическая величина	Размерность	Проектируемая текстовая строка
Полная мощность	Вольт-ампер Киловольт-ампер Мегавольт Ампер Милливольт Ампер	VA kVA MVA mVA
Угловая скорость	1/секунду 1/минуту 1/час	1/s 1/min 1/h
Угол	Радиан Секунда Минута Градус ¹ Град ¹	rad " ' grad ngrad
Скорость	Метров/секунду Миллиметров/секунду Миллиметров/минуту Метров/минуту Километров/минуту Миллиметров/час Метров/час Километров/час	m/s mm/s mm/min m/min km/min mm/h m/h km/h
Расход (поток объема)	Кубометр/секунду Кубометр/минуту Кубометр/час Литр/секунду Литр/минуту Литр/час	m ³ /s m ³ /min m ³ /h l/s l/min l/h
Поток массы	Килограмм/секунду Грамм/секунду Тонна/секунду Грамм/минуту Килограмм/минуту Тонна/минуту Грамм/час Килограмм/час Тонна/час	kg/s g/s t/s g/min kg/min t/min g/h kg/h t/h
Крутящий момент	Ньютонометр Килоニュトンометр Меганьютонометр	Nm kNm MNm
Температура	Кельвин Градус Цельсия Градус Фаренгейта	K C F

¹ 100 град = 90 градусов

Физическая величина	Размерность	Проектируемая текстовая строка
Теплоемкость	Джоуль/килограмм Килоджоуль/килограмм Мега джауль/килограмм	J/kg kJ/kg MJ/kg
Электрическое напряжение	Вольт Киловольт Милливольт Микровольт	V kV mV uV
Электрический ток	Ампер Миллиампер Кило Ампер Микроампер	A mA kA uA
Электрическое сопротивление	Ом МиллиОм КилоОм МегОм	Ohm mOhm kOhm MOhm
Отношение	Процент	%
Абсолютная влажность	Грамм/килограмм	g/kg
Частота	Герц Килогерц Мегагерц Гигагерц	Hz kHz MHz GHz
Относительный крутящий момент	Ньютонометр/ампер	Nm/A
Ускорение	Метр/секунду ² Метр/секунду ³	m/s ² m/s ³

3.19.2 Параметрирование для прикладного модуля FM 458

3.19.2.1 Основные понятия

- EXM 448

Модуль расширения коммуникаций EXM 448 для прикладного модуля FM 458

- CBP2

COMBOARD/коммуникационный модуль для Profibus DP

- DRIVE ES или DRIVE Monitor

Программный пакет для проектирования/параметрирования приводов (комплектных микропроцессорных устройств управления приводами Siemens)

3.19.2.2 Принципы коммуникации

Прикладной модуль FM 458 проектируется в корзине SIMATIC S7-400 вместе с одним или двумя модулями расширения коммуникаций EXM 448. Для того чтобы обеспечить прием и передачу запросов параметров, в свободное место X02 необходимо установить опциональный модуль, например CBP2.

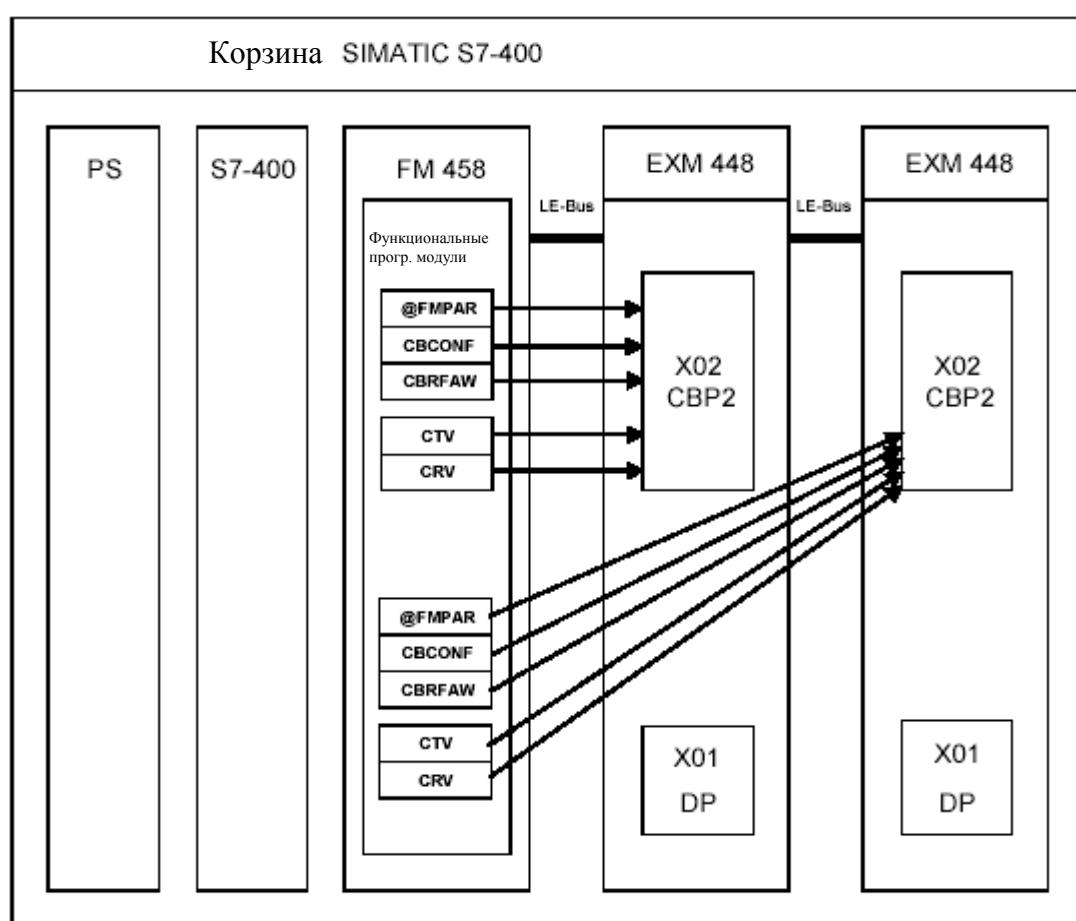


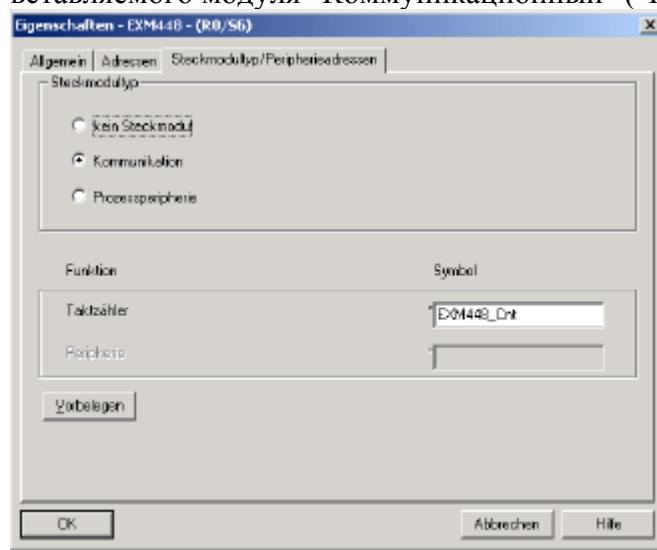
Рис. 3-97. Схематичное представление прикладного модуля FM 458 с двумя модулями расширения коммуникаций EXM 448.

3.19.2.3 Создание конфигурации аппаратного обеспечения (Hardwarekonfiguration)

Для параметрирования на модуле Simatic FM 458 необходимо следующее программное обеспечение:

- Корзина для Simatic S7-400
- Модуль источника питания для Simatic S7-400
- Центральный модуль (CPU) для Simatic S7-400
- Прикладной модуль FM 458 для Simatic S7-400
- Модуль расширения коммуникаций EXM 448

В HW-Konfig в свойствах EXM 448 должен быть активирован тип вставляемого модуля "Коммуникационный" ("Kommunikation").



- Коммуникационный модуль СВР2 (COMBOARD)

3.19.2.4 Объем функций

Для открытия возможности параметризации посредством Drive Monitor Вы должны спроектировать следующие функциональные (*программные*) модули:

- Центральный модуль @FMPAR
 - контролирует COMBOARD
 - обрабатывает запросы параметров
- Функциональный модуль СВCONF
 - предназначен для конфигурирования COMBOARD
 - служит для предоставления диагностических данных COMBOARD'a

Для дополнительных функций должны быть спроектированы следующие функциональные модули:

- Функциональный модуль CBRFAW

Для приема предупреждений от COMBOARD

- Функциональный модуль CRV

Функциональный модуль приема раздает значения интерфейсного блока данных по входам функциональных программных модулей CPU. С одной COMBOARD (например, СВР2) можно считать (или послать на неё) максимум только 16 PZD-слов.

- Функциональный модуль СТВ

Функциональный модуль объединяет (*в блок данных*) и передает только выходные значения функциональных программных модулей того CPU, для которого он был спроектирован.

3.19.2.5 Подключаемые служебные устройства

Для параметрирования прикладного модуля FM 458 вы можете использовать пакеты проектирования "DRIVE ES" или @Drive Monitor".

5 Шина PROFIBUS DP

Необходимые Аппаратные Средства и Программное Обеспечение

Для проектирования и эксплуатации шины PROFIBUS DP необходимы следующие программные и аппаратные средства:

- COM PROFIBUS
Заказывается дополнительно: заказной №.: 6ES5 895-6SE12 (Немецкая версия)
- DP плата для PC для загрузки COM базы данных посредством
COM PROFIBUS:
плата MPI или CP5411.

Характеристики

SIMADYN D имеет следующие характеристики на PROFIBUS DP:

Мастер: Коммуникационный субмодуль SS52 может работать на PROFIBUS DP как Мастер один или в связи с другими Мастерами при режиме работы "Мульти-Мастер".

Slave: Кроме функции Мастера имеется также функция Slave. Обе эти функции могут быть применены одновременно или раздельно.

Вход со множественным доступом: Каждый Slave на PROFIBUS DP подчинен только к одному Мастеру (Мастеру, в котором этот Slave был параметрирован) и, прежде всего, поддерживает связь только с этим Мастером. Функция "Вход со множественным доступом" разрешает чтение входных данных Slave'a и другим Мастерам. SIMADYN D поддерживает эту функцию и как Мастер и как Slave.

SYNC и FREEZE: При помощи служб (утилит) SYNC и FREEZE становятся возможными синхронизированные записи/чтение выходов/входов различных Slave'ов. SIMADYN D поддерживает эти службы (утилиты) как Мастер.

Длина Данных: Возможен обмен максимум 244 байта на канал или на Slave.

Время Передачи: При коротких телеграммах (до 32 байт) время передачи складывается исключительно из времени цикла SIMADYN (Abtastzeit) и времени цикла шины DP. При более длинных телеграммах должно быть прибавлено время аппаратной обработки модуля SS52 (max. 5 ms).

Согласованность: Данные внутри телеграммы всегда согласованы.

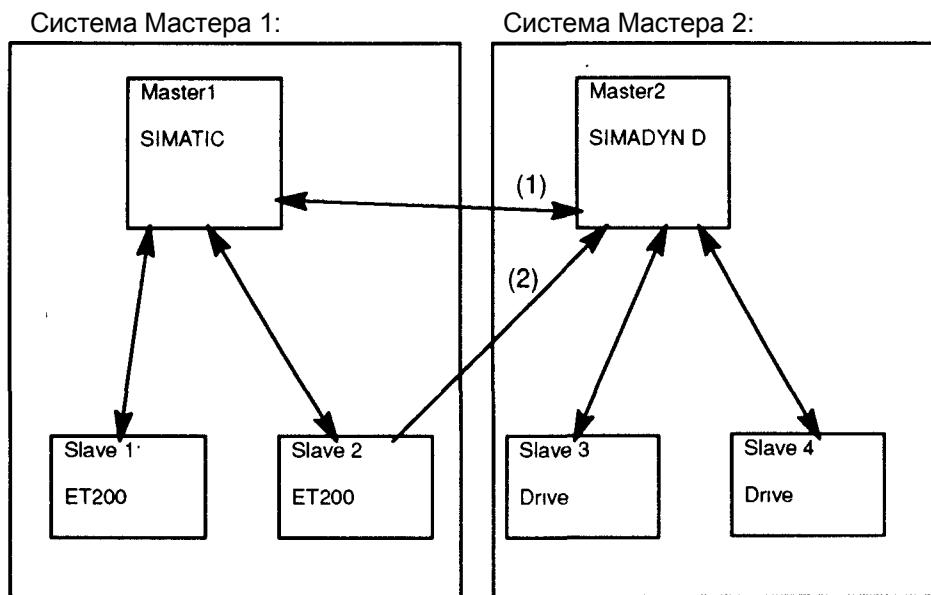


Рис 5.1-1 . Система Мульти – Мастер с функцией Slave'a (1) и входом со множественным доступом (2).

5.1 Проектирование в D7-SYS

Функциональные модули	Для связи PROFIBUS DP должны быть запроектированы следующие функциональные модули:
	<ul style="list-style-type: none"> центральный модуль связи @CSPRO максимум один функциональный модуль передачи и приема на Slave станцию может быть спроектирован максимум один функциональный блок синхронизации SYNPRO может быть спроектирован максимум один функциональный блок диагностики DIAL2A
Службы коммуникации	Разрешенные службы ¹ коммуникации: <ul style="list-style-type: none"> Процессорные данные Обработка параметров приводов с регулируемой скоростью
Режимы передачи	Разрешенные режимы передачи: <ul style="list-style-type: none"> Обновление (Refresh) Для приемников допускается мультиплексирование²

¹ Под "службой" понимается набор утилит и соглашений, обеспечивающих определенные доступ к данным и их передачу.

² Multiplexing – процесс совмещения нескольких сообщений, передаваемых одновременно в одной физической или логической среде.

5.1.1. Центральный модуль связи

Скорость передачи и адрес на шине PROFIBUS	Скорость передачи и адрес на шине PROFIBUS устанавливаются с одной стороны посредством CFC (функциональный модуль @CSPRO) и, с другой стороны, посредством COM PROFIBUS. Для корректной работы обеих параметров необходимо принять во внимание следующее: <ul style="list-style-type: none"> • Если COM – база данных еще не загружена <ul style="list-style-type: none"> — действительны определенные в CFC параметры; — субмодуль коммуникации SS52 находится в состоянии ожидания загрузки COM – базы данных. • Если COM – база данных загружена, а скорость передачи и адрес на шине COM – базы и проекта CFC согласованы <ul style="list-style-type: none"> — активизируется COM – база данных — субмодуль коммуникации SS52 начинает обмен пакетами данных. • Если COM – база данных загружена, но скорость передачи и адрес на шине COM – базы и проекта CFC не согласованы <ul style="list-style-type: none"> — действительны определенные в CFC параметры; — субмодуль коммуникации находится в состоянии ожидания загрузки. (Уже загруженая COM – база данных может быть активизирована, если значения скорости передачи и адреса на шине PROFIBUS, определенные на коннекторах центрального модуля связи, будут согласованы с соответствующими значениями, определенными при проектировании COM – базы данных (т.е. будут заданы точно такими же)).
---	--

5.1.2. Адресные коннекторы AT, AR

Задание значений на адресных коннекторах AT, AR	Особенности задания значений на адресных коннекторах AT, AR при использовании PROFIBUS DP:
--	--

Синтаксис задания величин:
"Имя канала.Первая часть адреса.Вторая часть адреса"

- **Имя канала**
 - Максимум 8 символов
 - ASCII-символы, исключая "точку" и @
 - Имена каналов всех функциональных блоков передачи и приема, определенных для конкретного коммуникационного субмодуля SS52, должны различаться (исключение – мультиплексный режим связи).
 - Имя канала не имеет специфического значения для PROFIBUS DP.
- Ввод имени канала завершается "." (точкой).

- **Первая часть адреса:**

- В первой части адреса определяется адрес Slave на PROFIBUS.
- Адрес Slave на PROFIBUS по каналу приема-передачи должен быть определен только один раз.
- Область значений: 0, 3 – 123.
- 0: означает, что данный канал сам является Slave-каналом и будет опрашиваться другим Мастером.
- 3 –123: одресация внешних Slave'ов.

- Ввод первой части адреса завершается точкой.

- **Вторая часть адреса:**

- Состоит максимум из двух символов.

- **1. Символ:** Расстановка байтов:

"1": Стандартная расстановка для PROFIBUS

Обмен данными проводится в формате "Моторола" (старший байт перед младшим байтом).

"0": Не стандартная (*дословно - исключительная*) расстановка

Обмен данными проводится в формате "Интел" (младший байт перед старшим байтом). Такая расстановка байтов может быть применена для партнеров по коммуникации, чье внутренне представление данных соответствует формату "Интел" (например SIMADYN D).

- **2. Символ:** Опция, только для приемника

"R":

Доступ осуществляется с блокировкой второго Мастера. Задание "R" допускается только для каналов приема ("вход со множественным доступом").

Если второй символ не задан, доступ к Slave проводит Мастер, в котором этот Slave был запараметрирован.

**Примеры задания
значений на
адресных
коннекторах**

- **AT – 'Sollwert.25.1'**

- Канал с именем **Sollwert** передает на **Slave** по адресу **25** на PROFIBUS. Обмен проводится в формате "Моторола".

- **AR – 'RECEIVE.117.0'**

- Канал с именем **RECEIVE** принимает от **Slave** по адресу **117** на PROFIBUS. Обмен проводится не стандартно - в формате "Интел".

- **AR – 'Eingang.33.1R'**

- Канал с именем **Eingang** (вход) принимает от **Slave** по адресу **33** на PROFIBUS с **блокировкой второго мастера**. Обмен проводится не стандартно - в формате "Интел".

- AT – 'Slaveist.0.1'
 - Канал с именем **Slaveist** передает как **Slave** для **Мастера - DP**.

5.1.3. Команды SYNC/FREEZE¹.

Общие положения	Команды SYNC и FREEZE синхронизируют вводы (запись) значений на группу Slave'ов и выводы (считывания) значений из нее. Функциональный модуль SYNPROM активизирует эти команды и поддерживает гарантию согласованности.
Согласованность	За обеспечение согласованности отвечает разработчик. При использовании команд SYNC/FREEZE речь идет о согласованности данных всех Slave'ов, принимающих участие в действии этих команд. Согласованность входных и выходных данных для одного Slave , естественно, гарантирована всегда.
SYNC	<p>При выборке команды SYNC DP – Мастер (SS52) ожидает в течение одного периода цикла шины, чтобы все Slave смогли сохранить обновленные выходные значения. Затем DP – Мастер посыпает для спроектированной группы Slave'ов широковещательную SYNC – телеграмму. В ответ все Slave этой группы одновременно актуализируют на своих выходах предварительно сохраненные значения.</p> <p>Гарантия согласованности: При проектировании необходимо обеспечить, чтобы в течение одного периода цикла шины после выборки команды SYNC выходные данные не подвергались изменению со стороны CPU SIMADYN D.</p>
FREEZE	<p>При выборке команды FREEZE DP – Мастер немедленно посыпает для спроектированной группы Slave'ов широковещательную FREEZE – телеграмму. Затем все Slave этой группы одновременно считывают значения на своих входах и сохраняют их в буферной памяти. По истечении одного периода цикла шины эти входные данные предоставлены для CPU SIMADYN.</p> <p>Гарантия согласованности: При проектировании необходимо обеспечить, чтобы в течение одного периода цикла шины после выборки команды FREEZE входные данные не обрабатывались (дословно – не оценивались) в CPU SIMSDYN.</p>

5.1.3. SYNC/FREEZE – варианты проектирования.

Общие положения	Ниже разъясняются некоторые термины и представляются различные варианты проектирования SYNC/FREEZE.
Термины	<ul style="list-style-type: none"> • Цикл шины Цикл, в течении которого DP – Мастер один раз опрашивает всех Slave'ов. Цикл шины проектируется и расчитывается в COM PROFIBUS посредством установки скорости передачи, числа и типов Slave'ов. Он (цикл шины) может быть выбран там как "Типовое время цикла данных" ("Typische Datenzykluszeit") в пункте меню Параметры шины (Busparameter).

¹ SYNC – от "синхронизация", FREEZE – от "замораживать"

- **Время опроса (Abtastzeit)**

Цикл, в котором производится обработка функционального блока SYNPRO и функциональных блоков приема – передачи (в CPU SIMADYN D).

Замечание

Цикл шины и время опроса независимы друг от друга.

- **Цикл синхронизации**

Цикл синхронизации проектируется посредством задания на коннекторе CNX функционального блока SYNPRO целых крат времени опроса.

(Цикл синхронизации = CNX * Время опроса).

Цикл синхронизации начинается всегда одновременно со временем опроса. SINC – команда активизируется функциональным модулем SYNPRO всегда в системном режиме в начале времени опроса.

Вариант конфигурации первый

Вариант конфигурирования 1 соответствует наиболее часто встречающемуся на практике:

- Генерируется SINC – команда
- Согласованность всех Slave'ов гарантирована
- Цикл синхронизации по крайней мере вдвое больше времени опроса (*Abtastzeit*) ($CNX > 1$).
 - Длина телеграммы, передаваемой каждому Slave'y (выхода), не должна превышать 32 байта.
 - Все блоки передачи и функциональный блок SYNPRO должны быть спроектированы с одним и тем же временем опроса (*давайте я буду дальше ругать "время опроса" по немецки – Abtastzeit*).
 - Функциональный блок SYNPRO должен быть спроектирован перед функциональными блоками передачи во временной последовательности обработки.
 - Выходной коннектор SOK функционального блока SYNPRO должен быть соединен с входными коннекторами Enable всех блоков передачи (на Slave относящейся к нему группы).
 - Цикл шины должен быть короче цикла синхронизации за минусом одного *Abtastzeit'a*. При выполнении программы проверьте, устанавливается ли выход SOK в "1" один раз в каждом цикле синхронизации, в противном случае время синхронизации необходимо увеличить.

Пример:

- Время синхронизации = $3 * Abtastzeit$
- Цикл шины = $2 * Abtastzeit$
- Предположим: функциональный блок SYNPRO обрабатывается в середине *Abtastzeit* (перед всеми блоками передачи)

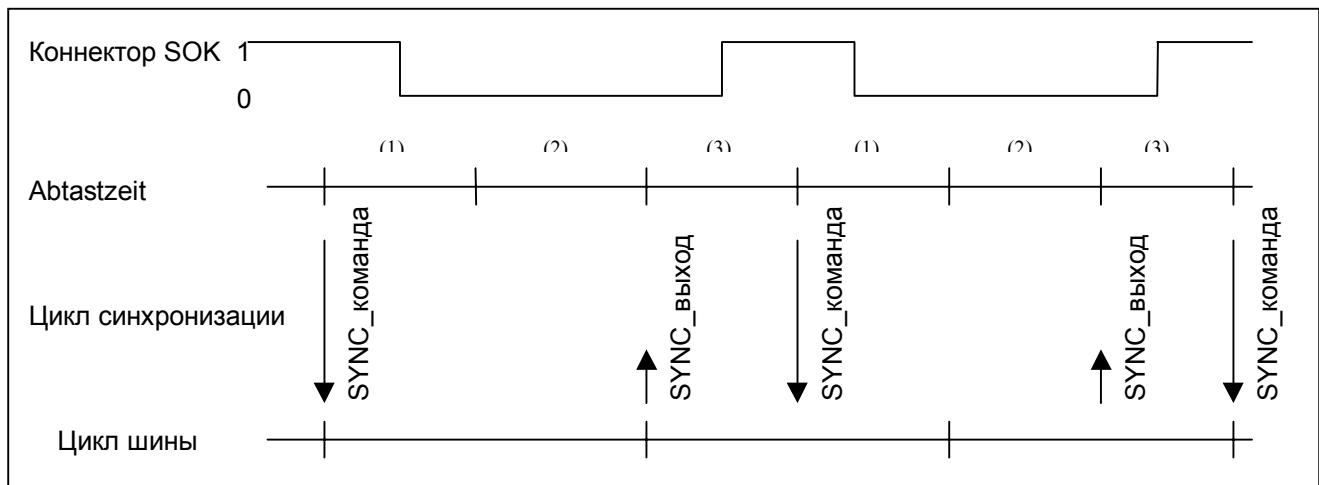


Рис 5.1-1 Временная диаграмма для первого варианта синхронизации.

После активизации SINC – команды¹ функциональные модули передачи блокированы ($SOK = 0$) в течение двух Abtastzeit (одного цикла шины). В третьем Abtastzeit после активизации SINC-команды модули передачи разблокируются ($SOK = 1$).

Вариант конфигурации второй

Вариант конфигурирования 2 соответствует наибольшей производительности SYNC:

- Генерируется SINC – команда
- Согласованность всех Slave'ов гарантирована
- Цикл синхронизации = Abtastzeit ($CNX = 1$)
 - Длина релеграммы, передаваемой каждому Slave'у (выхода), не должна превышать 32 байта.
 - Все блоки передачи и функциональный блок SYNPRO должны быть спроектированы с одним и тем же Abtastzeit.
 - Высокая скорость передачи ($>1,5$ Мбод). При более низкой скорости передачи временные условия не будут гарантировано соблюдены.
 - Время цикла шины должно составлять максимум половину Abtastzeit.
 - Кроме того, время цикла шины должно быть так мало, что смогло бы уложиться в промежуток от начала Abtastzeit до начала исполнения функционального блока SYNPRO. Это соглашение не может быть гарантировано, поэтому должно быть проверено во время функционирования программы.

Пример:

- Время синхронизации = Abtastzeit
- Цикл шины = $0,3 * Abtastzeit$
- Предположим: функциональный блок SYNPRO обрабатывается в середине Abtastzeit (перед всеми блоками передачи)

¹ Правильнее говорить об активизации функционального модуля SYNPRO, т.к. SYNC-команда активизируется только в начале Abtastzeit, что верно отражено на диаграмме. Теперь связь SYNPRO с модулями передачи (одинаковое Abtastzeit и строгая последовательность) не кажется лишней.

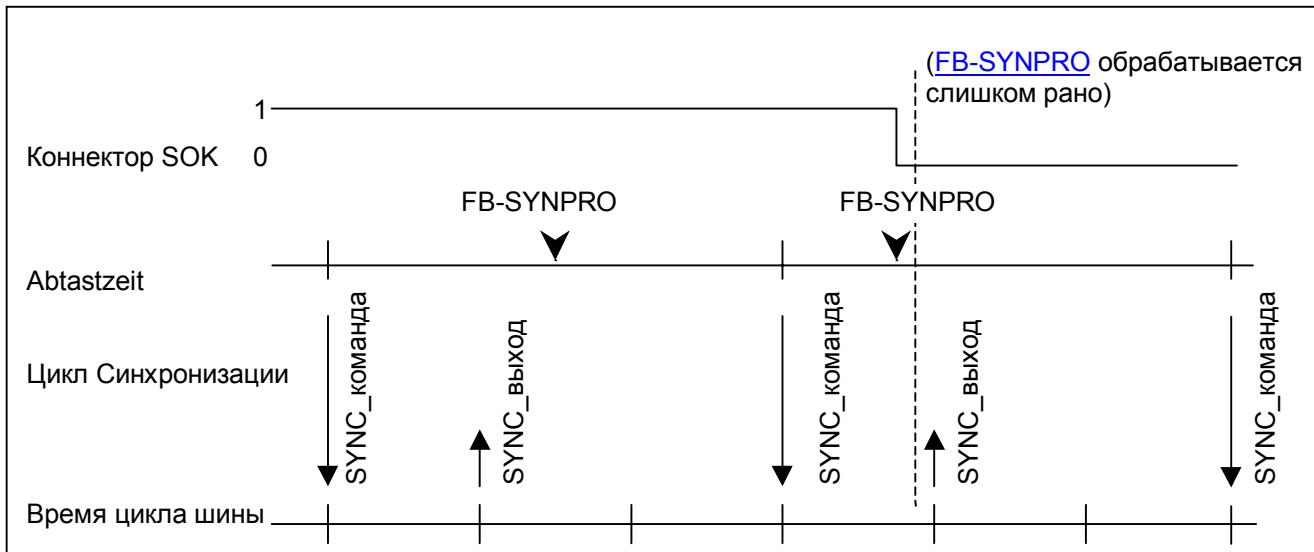


Рис 5.1-2 Временная диаграмма для второго варианта синхронизации.

Как правило блоки передачи всегда разблокированы ($SOK = 1$). Вследствии возможного смещения по времени точки начала расчета функционального модуля (далее *FB*) *SYNPRO*, передаваемые данные могут быть не актуализированы; в этом случае будут переданы значения предыдущего *Abtastzeit*. Цикл синхронизации и согласованность данных (цикл согласования данных) не будут при этом взаимосвязаны (не совпадут).

Пример работы корректно сконфигурированного синхроцикла (SYNC)

Наряду с маленьким временем синхроцикла возможно потребуется минимизация временных колебаний (сдвигов) в синхроцикле. Меры, обеспечивающие это:

- Избегайте нерегулярного трафика¹ данных по DP-шине; работайте с одним Мастером на шине; никаких временных подключений на станциях.
- В CPU SIMADYN, обеспечивающим связь с шиной, не должны быть сконфигурированы задачи, связанные с прерываниями по событиям тревоги.
- *Abtastzeit* ни в коем случае не должно перекрываться², это приводит к сбою *SINC*-команды или ее сдвигу на полный *Abtastzeit*.
- Определяйте при проектировании возможно большую скорость передачи и возможно меньшую длину телеграмм. (Время опроса Slave включается в плавающее временное смещение).
- Проектируйте *FB – SYNPRO* и принадлежащие ему *FB – передачи* в $T1 = T0$ (основное *Abtastzeit*). Синхрокоманда в этом случае будет генерироваться по прерыванию от основного такта. Это прерывание более строго привязано ко времени, чем прерывание, генерируемое в системном режиме (System-Mode ausgelöster Interrupt).

Вариант конфигурации третий

Вариант конфигурирования 3 соответствует немногим случаям практического применения *FREEZE*:

- Генерируются команды *SYNC* и *FREEZE* или одна *FREEZE*
- Согласованность всех Slave'ов гарантирована

¹ Трафик – рабочая нагрузка линии, мера объемов данных или сообщений, проходящих между пунктами в сети связи.

² Но тогда правильнее говорить об исключении **любых** задач по прерываниям, если эти прерывания не связаны с таймером, - PGMO, например, также должно исключить.

- Цикл синхронизации по крайней мере в три раза больше Abtastzeit (CNX > 3)
- Длины передаваемых каждому Slave или принимаемых от каждого Slave телеграмм, не должны превышать 32 байта.
- Все блоки передачи - приема и функциональный блок SYNPRO должны быть спроектированы в одном и том же Abtastzeit (в одном CPU).
- Функциональный блок SYNPRO должен быть спроектирован последним функциональным блоком во временной последовательности обработки.
- Выходной коннектор SOK функционального блока SYNPRO должен быть соединен с входными коннекторами Enable всех блоков передачи - приема (на Slave относящейся к нему группы).
- Цикл шины должен быть короче цикла синхронизации за минусом двух Abtastzeit. При выполнении программы проверьте, устанавливается ли выход SOK в "1" один раз в каждом цикле синхронизации, в противном случае время синхронизации необходимо увеличить.

Пример:

- Время синхронизации = $4 * \text{Abtastzeit}$
- Цикл шины = $2 * \text{Abtastzeit}$
- Предположим: функциональный блок SYNPRO обрабатывается в середине Abtastzeit (после всех блоков приема - передачи)

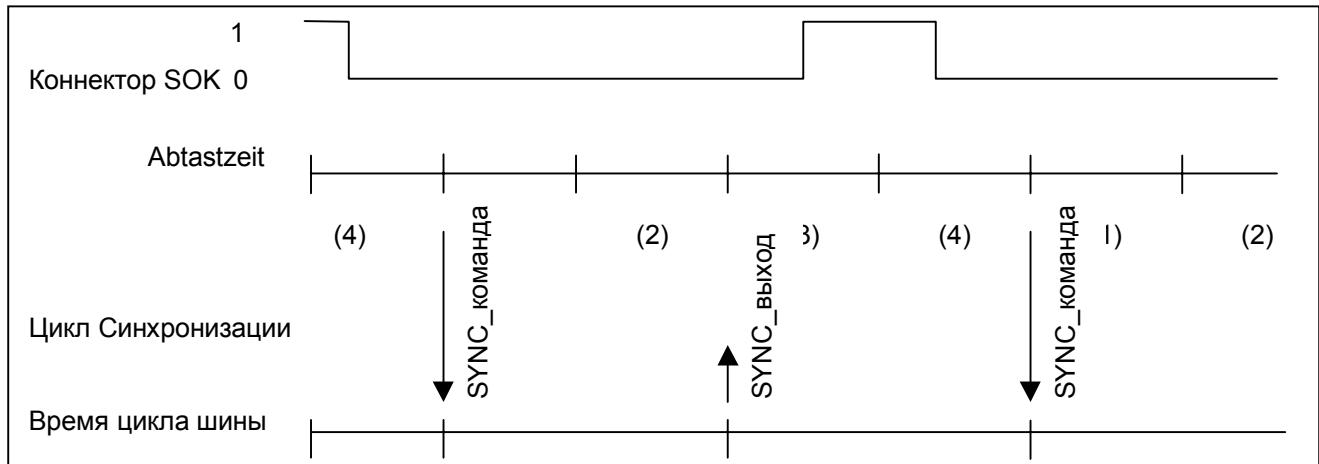


Рис 5.1-3. Временная диаграмма для третьего варианта синхронизации.

После активизации SINC-команды функциональные модули передачи-приема блокированы ($SOK = 0$) в течение трех *Abtastzeit* (один цикл шины + одно *Abtastzeit*). В четвертом *Abtastzeit* после активизации SINC-команды модули передачи разблокируются ($SOK = 1$).

5.1.4. Функциональный блок диагностики.

Общие положения При помощи функционального блока DIAPRO Мастер или Slave может провести специфическую диагностику шины PROFIBUS DP.

На входном коннекторе SEL выбираются диагностические данные (*тестовый блок данных?*), выдаваемые на шину. Они будут выданы на выходных коннекторах D01...D08.

Дальнейшая информация

Данные диагностики приводятся в документации пользователя "СОМ PROFIBUS" или в документации пользователя на отдельный Slave.

Обзор данных диагностики **SEL = 0:** нет диагностики

- Блок не выдает корректных диагностических данных.

SEL = 126: системная диагностика

- Системная диагностика дает обзор того, какие Slave'ы сообщили диагностические данные.
- 8 слов кодировано побитно.
- Каждый бит есть Slave с адресом на шине PROFIBUS, соответствующим приведенной ниже таблице.
- Если бит соответствующего Slave установлен – это означает, что данный Slave сообщает результат диагностики.

Коннектор	Bit 16	Bit 15	Bit 14	...	Bit 5	Bit 4	Bit 3	Bit 2	Bit1
D01	15	14	13	...	4	3	(2)	(1)	(0)
D02	31	30	29	...	20	19	18	17	16
...
D07	111	110	109	...	100	99	98	97	96
DOS	–	–	(125)	...	116	115	114	113	112

Таблица 5.1-1. Системная диагностика/протокол передачи данных – соответствие адресам Slave'ов на PROFIBUS.

SEL = 127: протокол передачи данных

- Протокол передачи данных дает обзор того, с какими Slave'ами состоялся обмен данными в пределах сконфигурированного времени (СОМ PROFIBUS).
- 8 слов кодировано побитно как при системной диагностике.
- Если бит соответствующего Slave установлен – это означает, что обмен данными с ним состоялся.

SEL = 128: Статус Мастера

- Выдается специфическая информация, относящаяся к Мастеру. (Для пользователя здесь имеет значение младший байт слова на коннекторе D01; хотя значения слов на других выходных коннекторах документированы, для них здесь пояснений не дается).

Выходной коннектор		Значение
D01	Младший байт	Состояние DP-Мастера: Останов (40h), Очистка (80h), Работа (C0h)
	Старший байт	Идент. № SS52 (старший байт) = 80h
D02	Младший байт	Идент. № SS52 (младший байт) = 37h
	Старший байт	(любое)
D03...D08		

Таблица 5.1-2. Специфическая информация, относящаяся к Мастеру.

SEL = 3 ... 123: Диагностика Slave

- Выдается диагностическая информация, относящаяся к Slave.
- Значение на коннекторе SEL соответствует адресу Slave на PROFIBUS.
- Диагностические данные зависят от типа Slave.
- Выдаются первые 16 байт диагностических данных Slave.
- Дальнейшая диагностика Slave проводится посредством установки на SEL значения >1000.

Дальнейшая информация

Специфические для Slave'a данные диагностики приводятся в документации пользователя "COM PROFIBUS" или в документации пользователя на конкретный Slave.

**Данные
диагностики для
SIEMENS Slave-DP**

Тип Slave'a		Общий SPC Slave	ET 200U	ET 200B	ET 200K	SPM Slave	ET 200C 8DE/8DA	Стандартные DP Slave'ы							
Коннектор															
D01	младший	Статус 1							Диагностика по стандарту "6 байт"						
	старший	Статус 2													
D02	младший	Статус 3							Специфич. диагност. устройства						
	старший	Номер Станции Мастера													
D03	младший	Идентификационный номер старшего байта													
	старший	Идентификационный номер младшего байта													
D04	младший	Заголовок блока диагностики, зарезервированного для устройства													
	старший	Диагностика устройства U	Диагн. устр. В	0	0	0									
D05	младший	Заголовок ID ¹ диагностики	0	0	0	0									
	старший	BG7-0	0	Канал 7-0											
D06	младший	BG15-8	0	Канал 15-8		0									
	старший	BG23-16	0	Канал 23-16		0									
D07	младший	BG 31-24	0	Канал 31-24		0									
	старший	Дополнительн. относящаяся к устройству диагностика	Не определено												
D08	младший		Не определено												
	старший		Не определено												

Таблица 5.1-3. Обзор форматов данных диагностики SIEMENS DP – Slave'ов.

¹ В немецком тексте – "kennungsbezogene" – дословно – "относящаяся к сигналам распознавания".

Формат слов
Статуса 1, 2 и 3

	Бит 8	Бит 7	Бит 6	Бит 5	Бит 4	Бит 3	Бит 2	Бит 1
Статус 1 (D01 младший байт)	S: Slave был параметри- рован другим Мастером	S: Сбой последней параметри- зирующей телеграммы	M: Сбой ответа Slave'a	S: Запро- шенная функция не поддержи- вается	S: Запись диагностики в специаль- ную диаг- ностичес- кую область	S: Конфи- гурация не корректна	S: Slave еще не готов для обмена данными	M: Slave не доступен на шине
Статус 2 (D01 старший байт)	M: Slave зарегистри- рован как "неактивный"	(Не исполь- зованн)	S: Slave получает команду Sync	S: Slave получает команду Freeze	S: Активен контроль доступа	S:1 (всегда)	S: данные диагностики должны быть считаны	S: Требуется проведение параметри- зации и конфигур. должны
Статус 3 (D02 младший байт)	S/M: Переданы не все данные диагностики	—	—	—	—	—	—	—

Таблица 5.1-4. Значения битов в Статусах 1, 2 и 3.

- **M:** диагностику распознает Мастер
- **S:** диагностику распознает Slave

Адрес Мастера на PROFIBUS

- Адрес на PROFIBUS того Мастера, который параметризовал данный Slave. Если Slave не параметризован – FFh.¹

Идентификационный номер

- Старший/младший байты:
Код для идентификации типа Slave'a.

Все другие диагностические данные являются специфичными для Slave'a.

В общем случае (стандартный DP – Slave) различают следующие диагностические блоки: относящиеся к устройству, относящиеся к кодам опознавания, относящиеся к каналу. Не все Slave'ы должны содержать специфические блоки диагностики.

¹ Т.е. для непараметризованного Slave'a Мастер имеет "фиктивный" адрес FFh.

Каждому блоку предшествует байт заголовка. Бит 7 и бит 8 являются идентификаторами диагностического блока:

Биты 7, 8 байта заголовка	Значение
Биты 7, 8 = 00	Диагностика, относящаяся к устройству
Биты 7, 8 = 01	Диагностика, относящаяся к кодам опознавания
Биты 7, 8 = 10	Диагностика, относящаяся к каналу.

Таблица 5.1-5. Значения битов 7 и 8 байта заголовка.

Биты с 1-ого по 6-ой обозначают:

- При диагностике, относящейся к устройству или к кодам опознавания, - длину диагностического блока, включая байт заголовка; область значений 2 ... 63.
- При диагностике, относящейся к каналу, - идентификационный номер; область значений 0 ... 63.

Проведение дальнейшей диагностики Slave'a

- Посредством задания SEL=1002 ...SEL=1123¹ Slave выдает диагностические байты 17 ...23.

5.2 Проектирование посредством СОМ PROFIBUS

Общие положения

Пакет СОМ PROFIBUS (Windows) предназначен для параметрирования (возможно также параметрирование при помощи предыдущей версии 2.1 СОМ ET200). При помощи СОМ PROFIBUS Вы определяете:

- Число и конфигурацию присоединенных к шине PROFIBUS DP участников
- Скорость передачи
- Важные для работы шины PROFIBUS параметры

Специфичные для SIMADYN D условия применения СОМ PROFIBUS :

- Конфигурируйте коммуникационный субмодуль SS52 как станцию типа SIMADYN D (Familie "SIMADYN").
- Не задавайте входной и выходной адреса.
- По окончании параметрирования загружайте базу данных по шине DP в SS52 из пункта меню **Datei > Export > DP-Master**.
- Альтернативным способом загрузки является загрузка по RS232 из пункта меню **Service > Laden SIMADYN-Master**.

¹ Здесь 2 ... 123 – адрес Slave'a на PROFIBUS.

5.2.1. Согласование с конфигурированием в CFC.

Соглашения	Необходимо провести согласование конфигураций как указано ниже:
	<ul style="list-style-type: none"> Скорость передачи и собственный адрес на PROFIBUS должны совпадать. Каждый сконфигурированный в СОМ Slave должен быть представлен в CFC-проекте драйвером приема – передачи – функциональным блоком CRV/CTV. Согласование происходит по адресу PROFIBUS (первая часть адреса на адресном коннекторе). Длины входных (принимаемых) и выходных (передаваемых) блоков данных по каждому Slave должны совпадать¹.
Указания на ошибки и предупреждения	Соблюдение соглашений будет проверено. При несоблюдении выдаются указания на ошибки или предупреждения:
	<ul style="list-style-type: none"> Область коммуникационных ошибок (мигающее "С" на процессорном модуле), а также выходной коннектор YTS функционального блока CRV/CTV. Выходной коннектор ECO функционального блока @CSPRO.
Замечание	<p>Следующие соглашения не проверяются:</p> <p>Структуры пользовательских блоков данных партнеров по коммуникации должны совпадать.</p> <p>Несоблюдение приводит к неверной интерпретации данных (например, перестановка байт внутри слова данных) между партнерами по коммуникации.</p>
Структура данных	<p>В SIMADYN D структура данных определяется в CFC посредством конфигурирования виртуальных связей (смотри главу "Коммуникационная служба "Процессорные данные"").</p> <ul style="list-style-type: none"> Для большинства Slave'ов PROFIBUS структура данных задается в СОМ PROFIBUS посредством задания идентификаторов в окне "Конфигурирование" ("Konfigurieren").

5.2.2. SS52 как PROFIBUS-Slave.

Проектирование	Коммуникационный субмодуль SS52 может быть запроектирован как чистый Slave или в комбинации Master – Slave.
	<ul style="list-style-type: none"> SS52 как чистый Slave обходится без СОМ – проектирования: На входной коннектор SLA функционального блока @CSPRO задается 1 или 2. Наряду с этим проектируются функциональные блоки CRV и/или CTV. В первой части адреса на коннекторах AR/AT задается "0".

¹ Имеется ввиду не равная длина передаваемых-принимаемых данных, а совпадение объявленных в CFC и СОМ длин.

- SS52 скомбинированный как Master и Slave
На входной коннектор SLA функционального блока @CSPRO задается "0" (предустановка).
 - Проектирование шины проводится посредством СОМ PROFIBUS. Каждому мастеру PROFIBUS поставлена в соответствие одна база данных ("Мастер-система"). Эти базы данных должны загружаться на соответствующие Мастера.
 - Если Мастер проектируется другим пакетом, то при проектировании SS52-Slave нужно спроектировать в СОМ PROFIBUS фиктивного Мастера. В этом случае обращайте внимание на корректную установку параметров шины: рекомендуется увеличить число активных станций (в базе данных) а также время цикла (Token – Rotation – Time).

5.2.3. Загрузка базы данных.

Варианты

Имеется два варианта загрузки базы данных:

Загрузка через PROFIBUS DP

- Загрузка через PROFIBUS DP является наиболее предпочтительным вариантом. Конечно, необходимо учитывать ограничения.
- Необходимы DP-совместимые РС-платы: CP5411, CP5511 или MPI-плата.
- Известные на текущий момент платы:
 - CP5411-ISA-плата (заказной номер: 6GK 1541-1A00).
Ограничения: в BIOS РС должно быть включено ограничение 16 Mbyte RAM. Вначале проверьте, пожалуйста.
 - MPI-ISA-плата (заказной номер: 6GK1514 – 10AA0).
Ограничения: максимальная скорость передачи при загрузке 500Кбод.
- Кроме того, имеются интегрированный в SIMATIC-PG (программатор) интерфейс MPI, а также CP5511 с разъемом PCMCIA.
- Загрузка через CP5412 невозможна.
- Соответствующие РС-платам драйверы инсталлируются вместе с СОМ PROFIBUS. Загрузка осуществляется из пункта меню COM PROFIBUS **Datei – Export –DP-Master**.

Загрузка через RS232

- При помощи программы "SS52LOAD" возможна загрузка в модуль SS52 созданной СОМ PROFIBUS базы данных в виде бинарного файла.
- SS52LOAD интегрирована в СОМ PROFIBUS (с версии 3.1).

- **Ограничения:**
Для обеспечения возможности загрузки в случае, когда спроектирован функциональный блок синхронизации SYNPROM, синхронный режим работы должен быть отключен (разрешающий коннектор EN = 0).
- Создание бинарного файла (*.2bf) проводится в СОМ PROFIBUS посредством выбора пункта меню **Datei > Export > Binaerdatei**.
- Загрузка проводится посредством SS52LOAD выбором пункта меню **Service > Laden SIMADYN-Master**.
- Интерфейс RS232 совместно с интерфейсом PROFIBUS выведен на 9-полюсный штекер модуля SS52. Кабель для связи с СОМ-портом РС изготавливается самостоительно. Выводы RS232 на SS25 (не стандартно):
 - 2 – Txd
 - 7 – RxD



ОСТОРОЖНО

Не перепутайте выводы RS232.

5.3 Ввод в работу/Диагностика

5.3.1. Светодиоды.

Желтый светодиод	В противоположность коммуникационным модулям желтый светодиод коммуникационного субмодуля SS52 не информируют непосредственно об активности шины.
Зеленый светодиод	Зеленый светодиод дает общую справку по коммуникационному субмодулю SS52 и по синхронизации с функциональным модулем @CSPRO SYMADYN D. Желтый светодиод дает справку по шине DP в связи с СОМ – базой данных.

Светодиод	Зеленый	Желтый
aus (не горит)	Процессор не работает	Шина не активна (во время запуска).
Быстрое мигание (0.2 с)	Серьезная ошибка. • Класс ошибки и код должны быть получены из FB-@CSPRO и сообщены Siemens AG.	Ошибка на шине (например, короткое замыкание). • Проверьте кабель и других клиентов шины.
Мигание (1 с)	Ожидание синхронизации с процессорным модулем SIMADYN. • Проверьте конфигурацию FB-@CSPRO.	СОМ база данных не существует или не активизирована (также во время загрузки). • Загрузите базу данных.
Медленное мигание (2 с)	—	Конфигурация STRUC и СОМ не согласована на 100%. Возможна ограниченная работа шины. • Согласуйте конфигурацию STRUC и СОМ проектов.
an (горит)	Модуль SS52 и синхронизация процессорами SIMADYN в порядке.	Нормальная работа шины с активированной СОМ-базой данных.

Таблица 5.3-1. Значения светодиодов коммуникационного субмодуля SS52.

Состояние при включении

- В первый момент после включения напряжения оба светодиода кратковременно включаются и выключаются.
- В течение инициализации (примерно 5 секунд) горит только зеленый светодиод.
- По истечении времени инициализации в нормальном случае включается также и желтый светодиод.
- После Reset'a оба светодиода остаются в своем последнем состоянии до тех пор, пока программное обеспечение не обновит управление светодиодами.

Состояние при загрузке

- При загрузке мигает желтый светодиод. (При высокой скорости передачи это состояние очень непродолжительно).
- Затем состояние как при включении.

Светодиоды не контролируют присоединение к шине и правильность параметризации всех Slave'ов. Если обмен данными с конкретным Slave'ом не в порядке, это можно идентифицировать по коду-"Break" относящегося к этому Slave'у функционального модуля (YEV = 0x0002 или YTS = 0x6014). Справка о текущем состоянии конкретного Slave'a содержится в диагностическом функциональном модуле DIAPRO.

5.3.2. Класс ошибки (ECL) и код ошибки (ECO).

Значения выходных коннекторов ECL, ECO функционального модуля @CSPRO

Выходные коннекторы ECL, ECO

- **Error-Class=0:** Имеется предупреждение. Предупреждение может быть частично устранено без Reset'a SIMADYN D. При большом числе предшествующих предупреждений отображается номер последнего.

- **Error-Class>0:** Имеется ошибка. Функциональный модуль @CSPRO устанавливает коммуникационную ошибку (на процессорном модуле мигает "С"). Для устранения ошибки устройство SIMADYN D должно быть сброшено (Reset).

Класс ошибки	Код ошибки	Значение
0 (Предупреждение)	0	Всё в порядке (О.К.)
	1	<p>База данных СОМ существует, но не активизирована, поскольку скорость передачи и номер станции не соответствует установленным на коннекторах BDR и МАА значениям.</p> <ul style="list-style-type: none"> Скорость передачи и номер станции должны соответствовать друг другу в СFC и СОМ проектах.
	2	<p>СОМ-база данных не существует.</p> <ul style="list-style-type: none"> Загрузите базу данных.
	3	Продолжается загрузка СОМ- базы данных с последующим стартом.
	4	<p>В СFC отсутствует конфигурация каналов участников-DP, спроектированных в СОМ-базе данных. Это состояние может появляться временно, после старта SIMADYN'a. Клиенты DP будут недоступны.</p> <ul style="list-style-type: none"> Согласуйте конфигурации СFC и СОМ.
	5	(Не используется).
	6	<p>Существует по крайней мере один спроектированный в СFC канал, не согласованный с СОМ-базой данных. Соответствующий функциональный блок SIMADYN'a генерирует ошибку связи (мигание "С").</p> <ul style="list-style-type: none"> Согласуйте конфигурации СFC и СОМ.
	7	<p>Существует по крайней мере один спроектированный в СFC канал, в принципе не согласованный с СОМ-базой данных. Соответствующий функциональный блок SIMADYN'a генерирует ошибку связи (мигание "С").</p> <ul style="list-style-type: none"> Скорректируйте проект СFC.
	8	<p>Не достаточно ресурсов. Не все каналы СFC будут обработаны.</p> <ul style="list-style-type: none"> Уменьшите проект СFC (каналы связи).
	9	<p>Существуют два канала, которые передают или хотят принимать от одного и того же Slave'a. Функциональный блок SIMADYN D, назначенный каналу, опрашиваемому более поздним по времени, генерирует ошибку связи (мигание "С").</p> <ul style="list-style-type: none"> Скорректируйте проект СFC.
	10	<p>Временный сбой в работе шины.</p> <ul style="list-style-type: none"> Проверьте кабель и клиентов шины.
>1 (Внутренняя ошибка)	(любой)	<ul style="list-style-type: none"> Запишите класс ошибки и код ошибки и сообщите Siemens AG.

Таблица 5.3-2. Значения класса ошибки и кода ошибки.

5.4 Пример прикладной программы связи PROFIBUS DP

Общие положения Пример прикладной программы описывает образец конфигурации состоящий из:

- SIMOVERT 6SE70
- ET200U
- ET200B
- SIMATIC S5

Предполагается знание проектирования SIMADYN D и языка программирования CFC.

Замечание Подробно будут разъясняться только те действия, которые имеют значение для выбранного образца конфигурации. Варианты или дополнительные компоненты будут отмечены, но детально не будут разъяснены. В тексте эти места помечаются отображением символа справа.

Следующие тематические области будут затронуты в данном примере прикладной программы:

- **Образец конфигурации**

Описание типичной для SIMADYN D конфигурации на PROFIBUS DP с соответствующими системными компонентами.

- **Проектирование под CFC**

Составление специфических модулей PROFIBUS DP и их проектирование в образцовой конфигурации.

- **Конфигурирование коммуникационного субмодуля SS52...**

Конфигурирование коммуникационного субмодуля SS52 посредством пакета параметризации COM PROFIBUS 3.0 и загрузчиком "SS52load".

5.4.1 Пример конфигурирования и системные компоненты

Общие положения

В качестве образцовой конфигурации выбраны следующие системы и устройства, при этом указанные адреса на PROFIBUS определены произвольно:

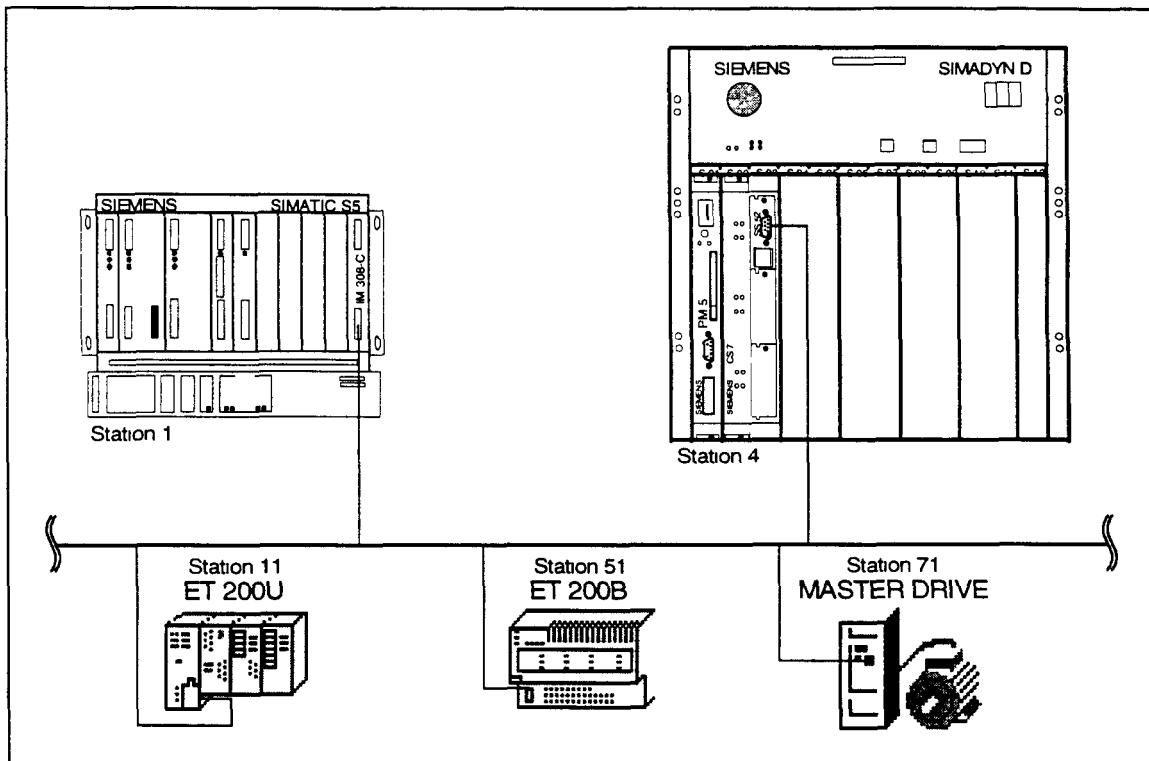


Рис 5.4-1 Пример конфигурации

Коммуникационные партнеры

Коммуникационными партнерами SIMADYN D являются в частности:

- **SIMATIC S5-105U (станция 1) – Мастер для SIMADYN D:**
SS52 подчинен Мастеру (S5), от которого ведется SIMADYN D. Обмен данными (число и формат процессорных данных) между обеими контроллерами может быть оформлен в произвольной форме. Будет определено (в примере):
 - S5 => SIMADYN D: три слова (ввод/вывод данных), одно слово (ввод данных), один байт (ввод данных), один байт (ввод данных).
 - SIMADYN D => S5: три слова (ввод/вывод данных), одно слово (вывод данных), один байт (вывод данных).

- **SIMOVERT MASTER DRIVE с CB1 (станция 71) как Slave:**

Для обмена данными с этим участником предоставлены в распоряжение пять определенных PRO-типов.

PRO: структура объекта "параметры – процессорные данные" (Parameter-Prozeßdaten-ObjekteStruktur) блока данных пользователя для регулируемых по скорости приводов. Имеется блок данных пользователя, состоящий либо из Значений – Идентификаторов – Параметров (PKW) и процессорных данных (PZD) (PRO-типы 1, 2, 5), либо только из процессорных данных (PRO-типы 3, 4). В примере конфигурирования проектируется PRO-тип 3. При этом два слова (слово управления и главное заданное значение) посылаются и два слова (слово состояния и главное истинное значение) принимаются.

- **ET 200 В (станция 51) как Slave:**

При применении этого типа Slave необходим точный выбор типа, посредством которого обмен данными будет определен автоматически. При типе 8DI/8DO один байт данных будет выдаваться (*записываться в устройство*) и один байт считываться (*из устройства*).

- **ET 200 U (станция 11) как Slave:**

При данной конфигурации ET 200 U (три модуля дискретных выходов и один модуль дискретных входов) три байта будут выдаваться (*записываться в устройство*) и один байт считываться (*из устройства*).

5.4.2 Список требуемых для SIMADYN D Hard- и Soft компонентов

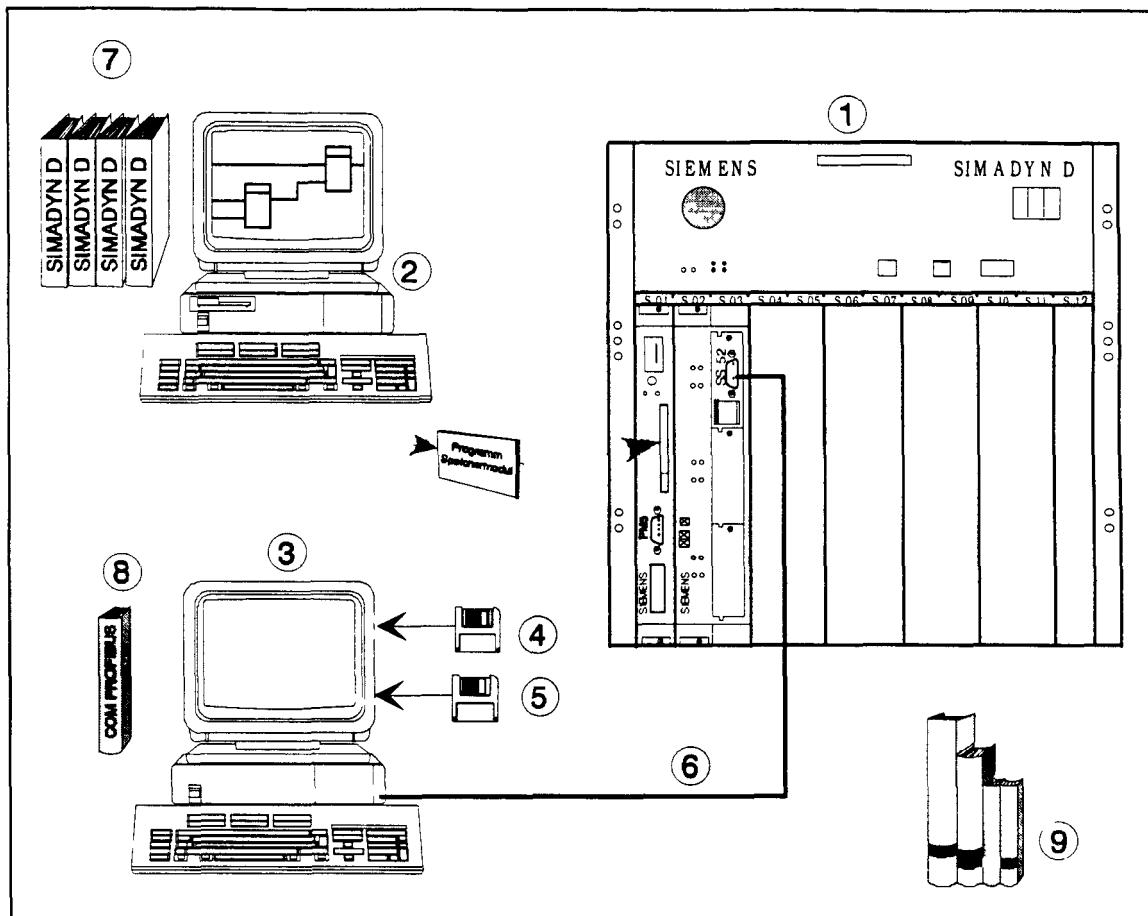


Рис 5.4-2 Hard- и Soft компоненты для SIMADYN D.

Легенда

1	Устройство SIMADYN D состоит, по меньшей мере, из: Носителя модулей, процессорного модуля, субмодуля памяти программы, коммуникационного модуля CS7 и субмодуля коммуникации SS52.
2	Устройство для проектирования в CFC: PC с Windows 95/NT как рабочая система, пакет STEP7, опциональный пакет D7-SYS и слот PCMCIA.
3	PC для работы с "COM PROFIBUS" и "SS52load" (может быть та же самая PC, что и для CFC) с: 3,5" дисководом, доступным для пользования последовательным интерфейсом, операционной системой Windows 3.1x или Windows 95.
4	Пакет параметрирования "COM PROFIBUS 3.0": Программа для конфигурирования шины PROFIBUS DP
5	Загрузчик "SS52load": Пакет для переноса созданной в "COM PROFIBUS" конфигурации DP в SS52 через COM-порт (RS232) PC.

6	Шнур RS232: Связь между SS52 (на 9-полюсный штекер SS52 кроме RS485 для PROFIBUS присоединен также RS232: 2-TxD; 7-RxD) и COM-портом PC (RS232). Этот кабель изготавливается по образцу (см. главу о загрузке СОМ-базы данных в SS52), т.к. RS232 SS52 нестандартен! При наличии коммуникационного процессора CP 5411 (или при наличии соответствующей вставной карточки в PC), в случае проведения загрузки через шину (RS485), отпадает необходимость в пакете "SS52load" и шнуре RS232. Однако CP 5411 не является составной частью данного руководства. Дополнительная литература (для не описанных здесь случаев или дальнейшего практического развития!):
7	Документация пользователя SIMSDYN D
8	Руководство по программному пакету СОМ PROFIBUS.
9	Руководства на остальных участников шины: SIMATIC S5, ET 200U, SIMOVERT Master Drives.

Таблица 5.4-1. Легенда Hard- и Soft компонентов для SIMADYN D.

5.4.3 Проектирование под STEP 7 CFC

Общие положения

Для облегчения текущего проектирования "Связи PROFIBUS DP" под CFC выстроим резюме специфических CFC-модулей и применяемому к ним синтаксису.

При проектировании коммуникационного субмодуля SS52 под CFC принимайте во внимание:

- На коммуникационный субмодуль SS52 ровно один центральный модуль @CSPRO.
- На партнер по коммуникации максимум один модуль передачи и/или приема.
- Допустимые службы коммуникации:
 - Процессорные данные
 - Обработка параметров для регулируемых по скорости приводов
- Допустимый режим обмена данными: обновление (Refresh) (для приемников также мультиплексный).
- На коммуникационный субмодуль SS52 максимум один модуль синхронизации SYNPRO.
- На коммуникационный субмодуль SS52 максимум один модуль диагностики DIAPRO.

Функциональные модули

Центральный модуль связи PROFIBUS DP @CSPRO.

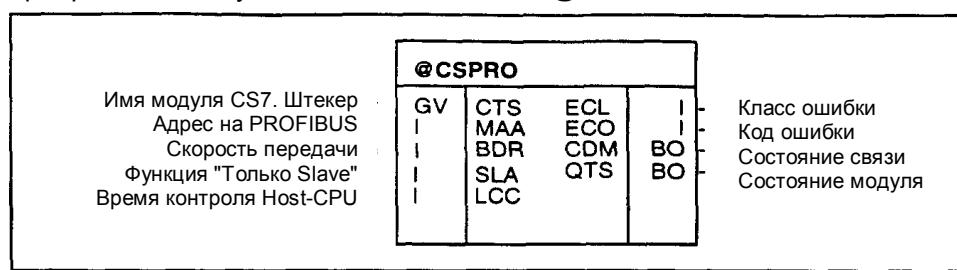


Рис. 5.4-3. Центральный модуль связи PROFIBUS DP @CSPRO.

- **Применение**

Этот функциональный модуль инициализирует и контролирует связь PROFIBUS DP (CS7 с установленным в него SS52). Он может быть спроектирован только в диапазоне времени опроса (Abtastzeit) $32ms \leq TA \leq 255ms$.

- **Коннектора**

Применительно к коннекторам ECL, ECO, CDM, QTS и YTS речь идет о сервисных и диагностических коннекторах, которые, как правило, используются при вводе в эксплуатацию SIMADYN D. В нашем проекте они не будут использоваться.

Дополнительную информацию

о коннекторах центрального модуля связи PROFIBUS DP смотри документацию пользователя "SIMADYN D, библиотека функциональных модулей".

CTS	На этот инициализирующий коннектор задается проектируемое имя модуля CS7 (идентичное объявленному в Мастер-программе, в нашем случае: D04) и идентификатор места вставки, где расположен SS52 (X01, X02 или X03, в нашем случае: X02).
MAA	Модулю SS52, как и всем абонентам шины, присваивается адрес станции. Он задается на этом коннекторе (число между 1 и 123, в нашем случае: 4).
BDR	На этом коннекторе устанавливается скорость передачи, с которой SS52 работает на шине. Значение задается в виде кода: 0=9,6 kBaud, 1=19,2 kBaud, 2=93,75 kBaud, 3=187,5 kBaud, 4=500 kBaud, 5=1,5 MBaud, 6=3 MBaud, 7=6 MBaud, 8=12 MBaud, (в нашем случае: 5)
SLA	Инициализирующий коннектор при выборе функции "Только Slave": 0: SS52 работает на PROFIBUS как Мастер и/или как Slave. Необходима загрузка СОМ-базы данных. 1 или 2: SS52 работает на PROFIBUS как чистый Slave без СОМ-базы данных PROFIBUS. 1: Slave либо только с входами, либо только с выходами. 2: Slave с входами и с выходами. (в нашем случае: 0).
LCC	Инициализирующий коннектор для времени, посредством которого субмодуль SS52 контролирует SIMADYN-Host-CPU: < 0: нет контроля 0...10: Время контроля = 1s (по умолчанию) >10: Время контроля в 1/10s (в нашем случае 0)

Таблица 5.4-2. Коннектора центрального модуля связи PROFIBUS DP.

5.4.4 Применение модулей передачи и приема

Общие положения Для PROFIBUS DP проектируются функциональные модули коммуникационной службы Процессорные Данные (Prozessdaten).

Адресные коннекторы AT и AR каждого модуля, поддерживающего интерфейс данных на SS52, должны удовлетворять следующим соглашениям:

AT/AR – ' Имя канала.Адресная часть 1.Адресная часть 2'

Имя канала

- Должно строго соответствовать общим правилам коммуникации (имена каналов всех модулей приема/передачи, связанных с данным коммуникационным субмодулем SS52, не должны совпадать).
- Должно состоять из не более 8 символов.
- Не иметь для PROFIBUS специфического значения.

Адресная часть 1

- В этой части адреса задается адрес партнера по коммуникации на PROFIBUS .
- Адрес 0 превращает канал в Slave-канал, который будет опрашиваться другим Мастером шины.
- По адресам 3...123 опрашиваются внешние Slave'ы.
- Конкретный PROFIBUS – адрес должен быть представлен по каналу приема/передачи только один раз.

Адресная часть 2

Эта часть адреса задается одной или двумя символами:

- **Первый символ:** Определение порядка байтов при обмене словными значениями для различных партнеров по коммуникации.
 - **1 = формат Motorola** (старший байт перед младшим байтом)
Чтобы формат телеграммы соответствовал нормам PROFIBUS, должен применяться стандартно для всех нормализованных абонентов шины, обменивающихся словными величинами (аналоговые входа/выхода, SIMOVERT, SIMATIC и т.п.).
 - **2 = формат Intel** (младший байт перед старшим байтом)
предназначен для обмена данными с устройствами, в которых обработка данных осуществляется как в SIMADYN D в формате Intel (например, второй SS52).

Партнер по коммуникации	Первый символ
SIMOVERT Master Drives с CB 1 (нормированный абонент шины)	1
Децентрализованная периферия ET200 (нормированный абонент шины)	1
SIMATIC (IM 308 C) (нормированный абонент шины)	1
SIMOREG 6RA24	1
MICRO / MIDI Master (нормированный абонент шины)	1
SIMADYN D (SS52) (Партнер по связи должен поддерживать установленные соглашения)	2 ¹

Таблица 5.4-3. Порядок байтов для различных партнеров по коммуникации.

- **Второй символ** (опция, только для приемников):
Заданием "R" на канале приема открывается право доступа к чтению этого канала другими Slave'ами (Shared input – вход со множественным доступом).

5.4.5 Пример конфигурирования в CFC

Общие положения

Здесь речь пойдет не столько об обработке процессорных данных, сколько, в первую очередь, о реализации объявленных коммуникационных связей к прочим абонентам шины.

CFC-план с пояснениями показывает проектирование PROFIBUS DP. CFC-план не претендует на охват всех тонкостей проектирования.

Нужно проектировать:

- Процессорный модуль PM5 на месте вставки S01 под именем D01_P1
- Коммуникационный модуль CS7 на месте вставки S02 под обозначением D02
- Субмодуль коммуникации SS52 (D042) в CS7, штекер X01.

¹ В немецком тексте, вероятно ошибочно, указан 0; или я чего то капитально не понял.

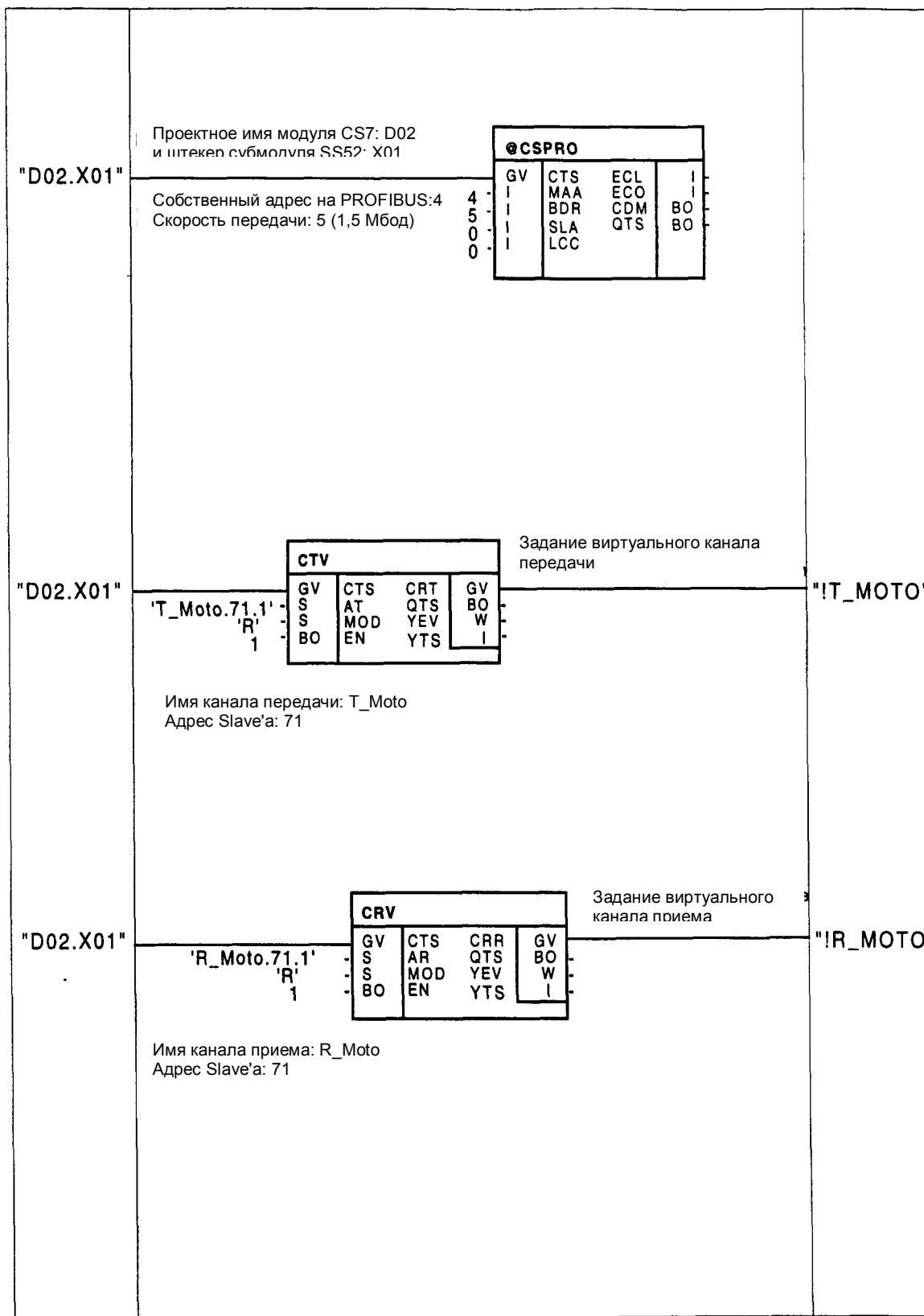


Рис. 5.4-4. CFC- план (часть 1) примера конфигурации.

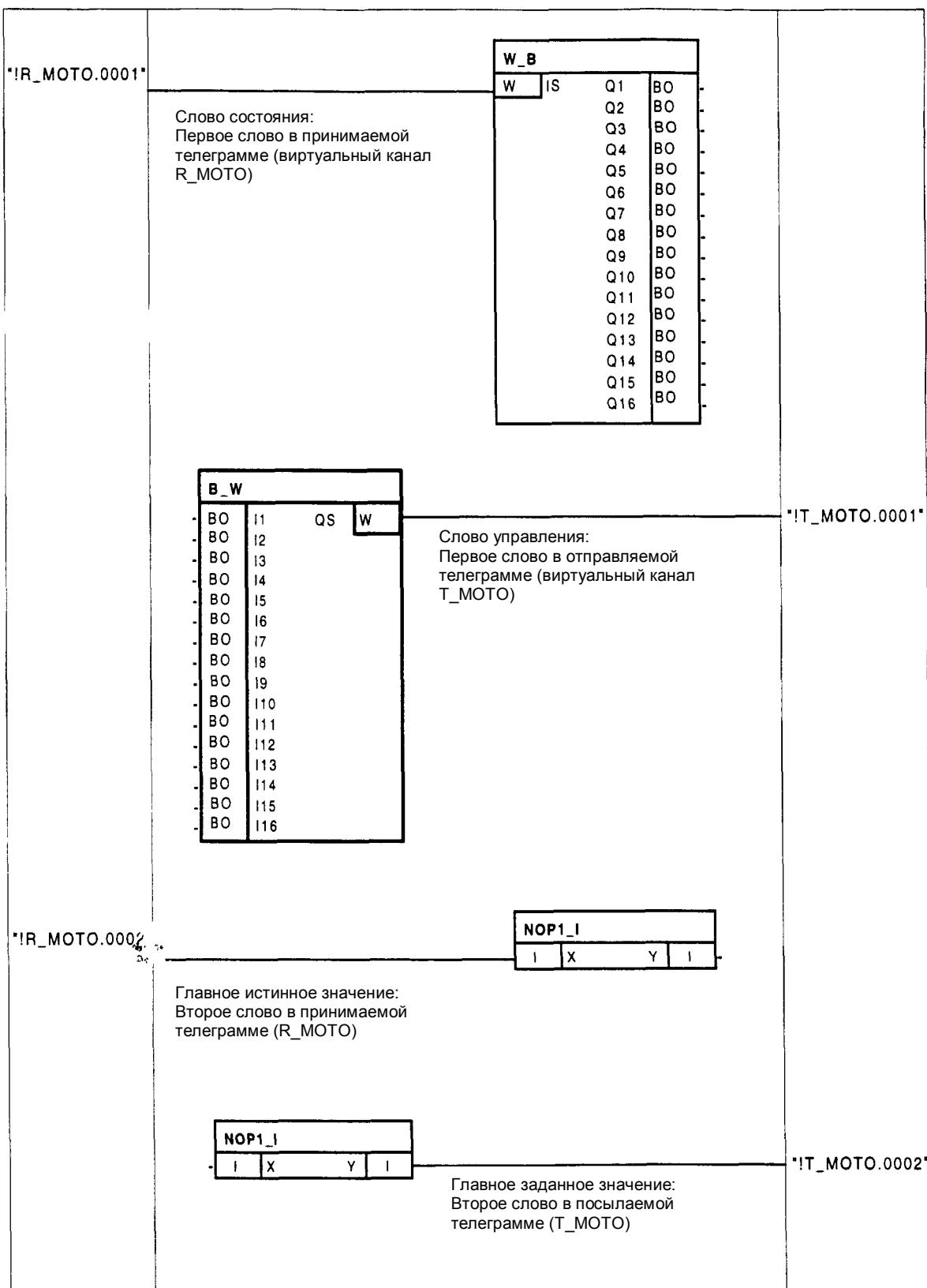


Рис. 5.4-5. CFC- план (часть 2) примера конфигурации.

5.4.6 Конфигурирование субмодуля коммуникации SS52 посредством COM PROFIBUS

Общие положения

Если на одном из трех мест CS7 устанавливается и проектируется субмодуль коммуникации SS52, изменяются объявления в части связи между штекером шины коммуникационного субмодуля SS52 и модулями приема/передачи. Т. к. SIMADYN D является свободно проектируемой системой, прежде всего должны быть созданы следующие логические структуры:

- Определение параметров шины (скорость связи, ...)
- Описание коммуникационных связей между абонентами (кто связан с кем и с какими функциями?)
- Определение коммуникационных объектов
(Коммуникационными объектами являются данные пользователя. Применительно к SIMADYN D они состоят из процессорных данных и данных устройств. В примере конфигурации коммуникация проводится только с процессорными данными – Prozessdaten.)

Эти данные (в дальнейшем обозначенные как СОМ-база данных) загружаются в жестко интегрированный в SS52 чип памяти, изменяются и корректируются посредством Downloads через 9-полюсный штекер.

5.4.7 Генерация СОМ – базы данных посредством COM PROFIBUS

Представление

Мастера и Slave'ы поддерживаемых в шинной системе коммуникационных партнеров проектируются в виде графических объектов пользователя и в виде списка.

Вначале, посредством выбора каждого абонента, задаются коммуникационные связи примера конфигурации.

Параметризация первого узла системы

1. После старта программы выбором пункта меню Datei > Neu организовывается первый Мастер системы.

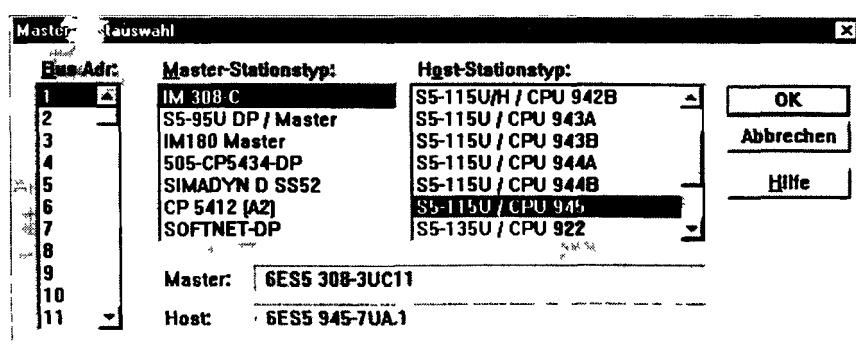


Рис. 5.4-6. Диалоговое поле "Выбор Мастера узла".

2. Запоминанием (**Datei > Speichern unter ...**) под любым именем (в нашем случае: "Пример" – "Muster") создается первый узел системы под идентификатором "Mastersystem <1>". При этом цифра (в нашем случае: 1) идентична установленному адресу на PROFIBUS. Первым шагом определяется, кто будет "разговаривать" с этим узлом системы.

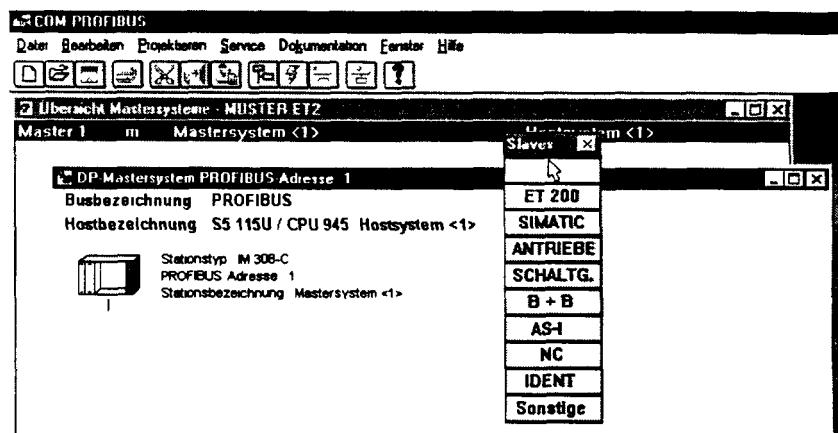


Рис. 5.4-7. Окно "Мастер-система – DP с PROFIBUS – адресом 1"

3. После активизации плоской включающей кнопочки "ET200" в меню "Slave" курсор мыши принимает форму пустого ящика с направленной вверх стрелкой.



Теперь можно назначить Slave'a для S5-станции, для чего курсор мыши позиционируется под символом станции и производится щелчок мышью.

4. После ответа на запрос PROFIBUS –адреса можно в следующем окне выбора выбрать партнера по коммуникации.

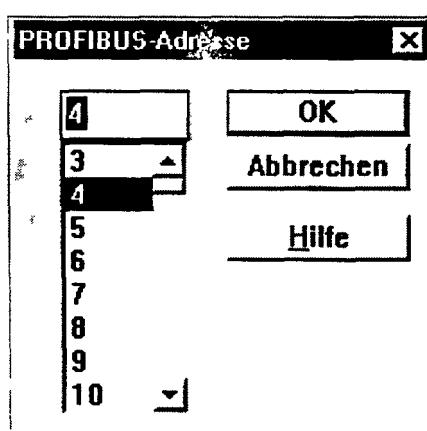


Рис. 5.4-8. Окно "PROFIBUS – адрес"

5. Весь диапазон набора установок в окне "Характеристики Slave'ов – "Slaveeigenschaften" для примера конфигурирования не имеет значения. Могут быть приняты стандартные установки. Важны только: группа (в нашем случае: SYMADYN), тип станции (в нашем случае: "SS52 Master/Slave") и плоская кнопочка "Конфигурировать... – Konfigurieren...".

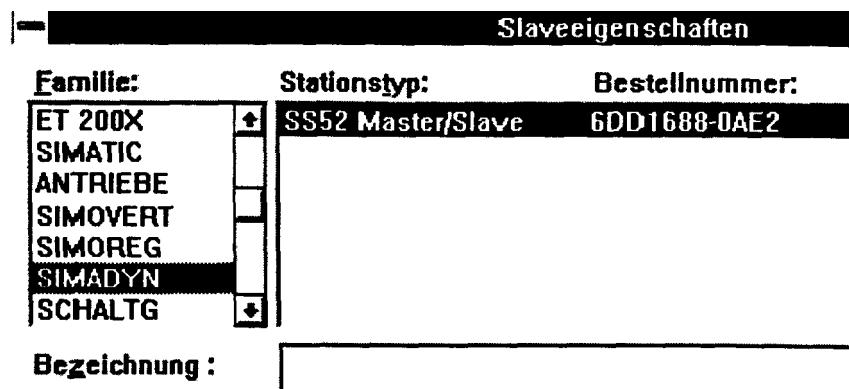


Рис. 5.4-9. Диалоговое поле "Характеристики Slave'ов - Slaveeigenschaften"

6. Перед началом конфигурирования необходимо сквитировать (OK) в диалоговом поле "Выбор Мастер-узла – Master-hostauswahl" заданные установки.

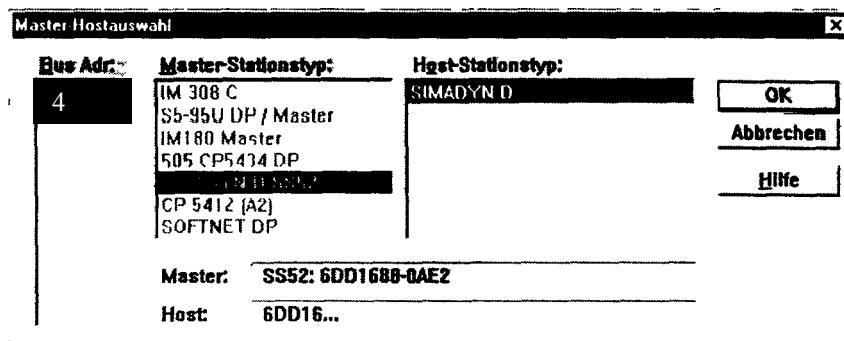


Рис. 5.4-10. Диалоговое поле "Выбор Мастер-узла – Master-hostauswahl "

7. После этого проводится конфигурация собственно абонента шины. Для коммуникационного субмодуля SS52 это окно конфигурирования первоначально совершенно пусто. Теперь нужно записать структуру данных пользователя в листе диалогового поля "Konfigurieren: SIMADYD D Slave..."

Примечание

В "Мастер-системе <1>" мастером является S5, так что работа передатчиков и приемников рассматривается с этой точки зрения (I/O адреса S5).

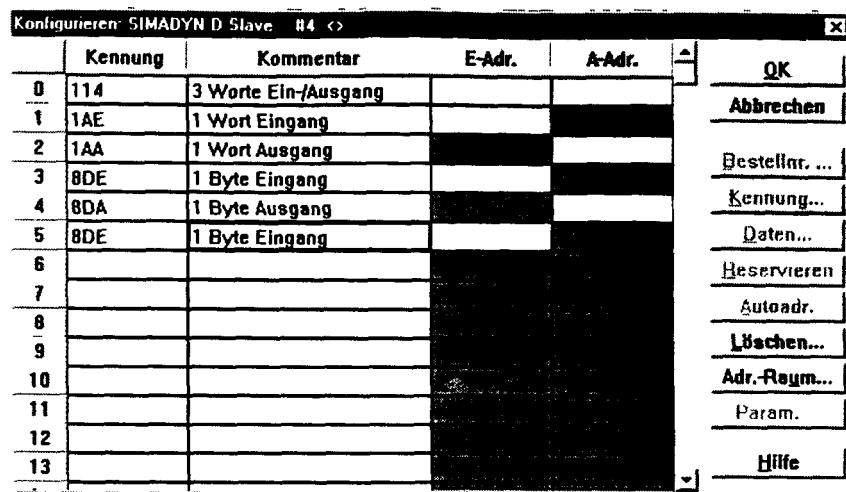


Рис 5.4-11. Диалоговое поле "Konfigurieren: SIMADYN D Slave..."

8. В колонке "Характеристика – Kennung" регистрируются все типы данных. Для этого активизируются соответствующие диалоговые окна. Это достигается посредством двойного щелчка на выбранной ячейке или посредством маркировки ячейки и последующего выбора плоской кнопки "Kennung...". Должны быть установлены следующие параметры:

- **Тип**

Выбор между:

- Вход, выход
- Вход/выход
- Пустое место
- Специальный формат

- **Длина**

От 1 до 16

- **Формат**

Выбор между словным или байтовым форматом

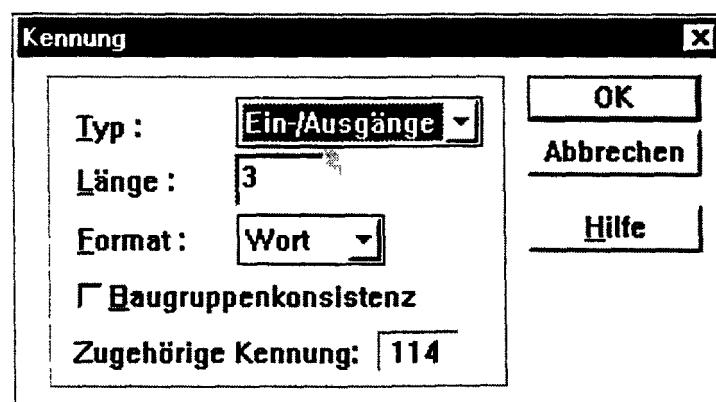


Рис 5.4-12 Диалоговое поле "Kennung".

9. Завершением диалога посредством OK соответствующие характеристики регистрируются в листе. Очередность процессорных данных в телеграмме определяется позицией записи характеристики соответствующих данных в области входных/выходных адресов (оставшиеся серыми поля не принимаются во внимание). Запись в колонки комментариев является опцией и оформляется произвольным образом. Установка адреса ("E-Adr" и "A-Adr") для базы данных SIMADYN D не требуется.

На этом заканчивается создание первого узла системы, в котором SIMADYN D подчинен S5 как Slave. Параметрирование для этой части теперь закончено. Обратите внимание, что до сих пор речь шла о конфигурационных данных применительно к IM308 (S5) и они не могут быть подвергнуты дальнейшей обработке, так как не имеют отношения к SS52.

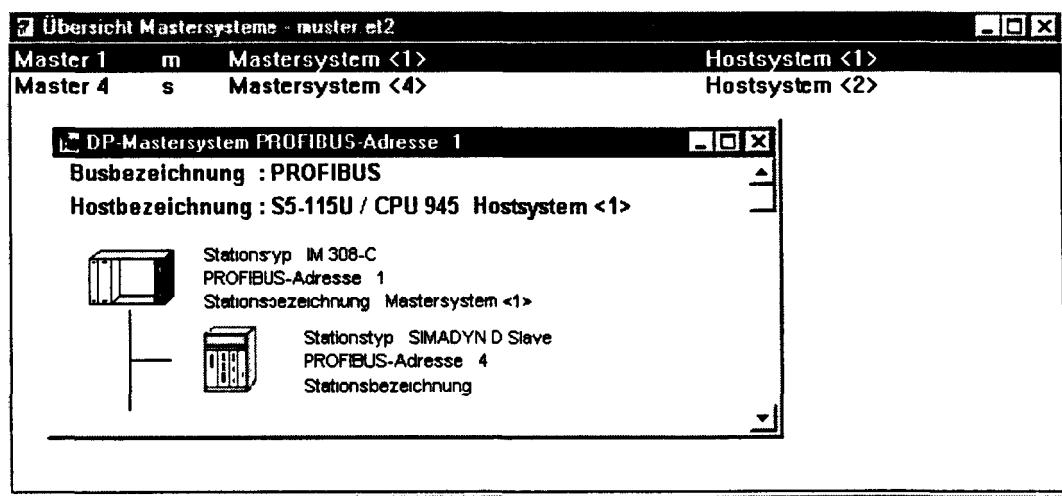


Bild 5.4 Рис 5.4-13. Окно "Мастер – система DP с PROFIBUS адресом 1"

**Параметризация
второго узла
системы**

1. Двойным щелчком на "Master 4" закрывается первый узел системы и делается доступным для параметризации SS52 – мастер второго узла.

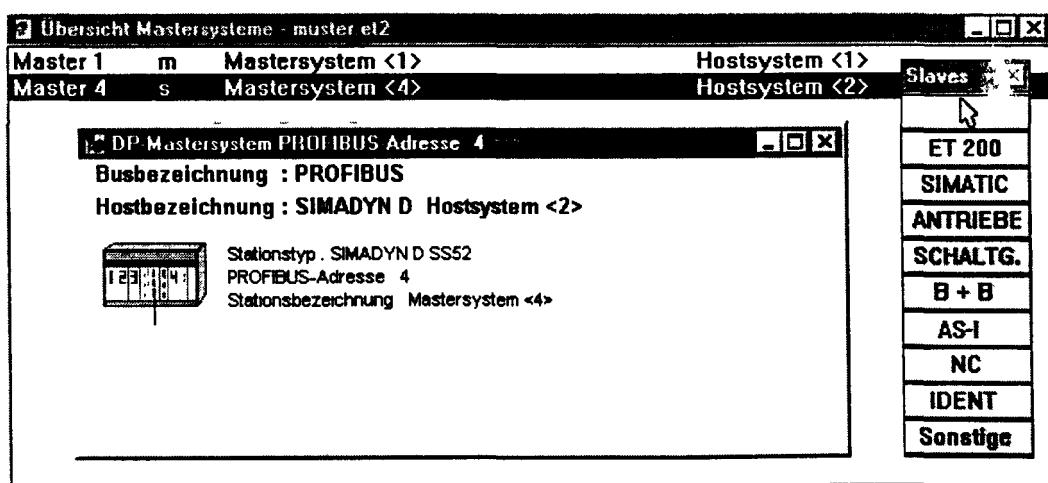


Bild 5.4 Рис 5.4-14. "Обзор Мастер-системы".

2. Как показывает следующий двойной щелчок на символе "SIMYND", было важно вначале описать SS52 как Slave первого узла системы ("Hostsystem <1>"). Уже сформированный формат телеграммы автоматически переносится в конфигурацию SS52, с тем отличием, что телеграмма меняет расположение данных в адресном пространстве: выхода S5 становятся входами для SIMADYN D и наоборот.

Характеристики данных (в меню "Konfigurieren...") теперь выделены серым цветом и для этой конкретной связи настоящего узла системы более не изменяются (характеристики и комментарии принадлежат S5). Установленные значения подтверждаются посредством "OK". Связь с S5 описана полностью.

Konfigurieren: SIMADYN D Slave #4 <>					
	Kennung	Kommentar	E-Adr.	A-Adr.	
0	114	3 Worte Ein-/Ausgang			
1	1AE	1 Wort Eingang			
2	1AA	1 Wort Ausgang			
3	8DE	1 Byte Eingang			
4	8DA	1 Byte Ausgang			
5	8DE	1 Byte Eingang			
6					
7					
8					
9					

Рис 5.4-14. Диалоговое поле "Konfigurieren: SIMADYN D Slave".

3. Теперь могут быть спроектированы функции Мастера для SS52. Для этого нужно вернуться к окну Мастер системы DP с PROFIBUS-адресом 4. После новой активизации меню Slave'ов (курсор мыши изменяется) начинаем **построение транзитной связи** от ET200U, ET200B к SIMOVERT Master Drive, причем после каждого вызова компонента запрашивается его адрес на PROFIBUS. При ответе на запрос автоматически раскрывается окно "Slaveeigenschaften" ("Свойства Slave'a"), в котором, как это уже описывалось, посредством **Konfigurieren...** производятся необходимые установки.

4. Так как по отношению к выносным (в *тексте - поле*вым) устройствам речь идет о чистых Slave'ах, для каждого из них возможно параметрирование в ограниченной форме в зависимости от их функций, устройства и "личного интеллекта". Отдельные конфигурации выглядят, как указано ниже:

— ET200U

Модульное построение с тремя выходными модулями (по 8 цифровых выходов) и одним входным модулем (8 цифровых выходов). В соответствии с этим должны три байта передаваться и один байт приниматься.

Konfigurieren: ET 200U #11 <>					
	Kennung	Bestellnummer	Kommentar	E-Adr.	A-Adr.
0	8DA		1 Byte Ausgang		
1	8DA		1 Byte Ausgang		
2	8DA		1 Byte Ausgang		
3	8DE		1 Byte Eingang		

Рис 5.4-16. Окно "Konfigurieren".

— **ET200B**

Компактное устройство с восемью цифровыми выходами и восемью цифровыми входами: По одному байту в передаваемой и принимаемой телеграмме. В соответствии с этим характеристика полностью определена выбором модуля.

Konfigurieren: B-8DI/8DO DP #51 <>				
	Kennung	Kommentar	E-Adr.	A-Adr.
0	8DA	1 Byte Ausgang		
1	8DE	1 Byte Eingang		

Рис 5.4-17. Окно "Konfigurieren".

— **SIMOVERT Master Drive**

Интеллектуальный Slave: применительно к описанию преобразователя допускаются пять различных форматов телеграмм (PRO-типов). Это должно быть уже определено при проектировании и здесь также не может быть изменено. (Ячейки маркированы серым цветом и деактивированы).

Konfigurieren: MASTER DRIVES CB1 #71 <>				
	Kennung	Kommentar	E-Adr.	A-Adr.
0	2AX	PZD 2 Words		

Рис 5.4-18. Окно "Konfigurieren".

5. По окончании конфигурирования дисплей должен выглядеть следующим образом:

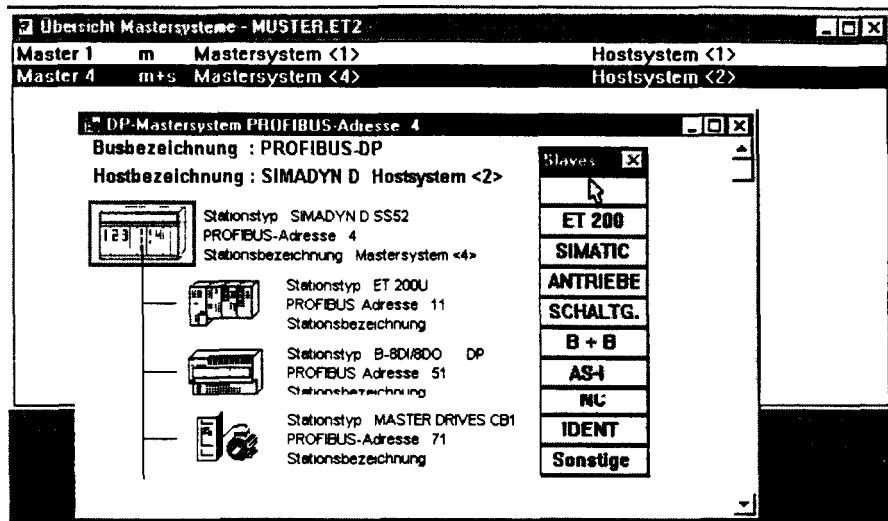


Рис 5.4-19. Окно "DP – Mastersystem PROFIBUS – Adresse".

Изменение конфигурации Slave'ов

Конфигурация каждого Slave'a может быть впоследствии изменена и скорректирована.

1. Любой символ на вышеприведенной иллюстрации может быть выбран двойным щелчком мыши. Из окна "Slaveeigenschaften" ("Свойства Slave'ов") можно вновь перейти к диалоговым окнам конфигурирования.
2. Для окончательного завершения параметрирования необходимо, в качестве последнего шага, установить параметры шины. Посредством **Projektieren > Busparameter...** (Проектирование > Параметры шины) открывается диалоговое окно, в котором для нашего примера конфигурирования имеют значение только профиль шины (PROFIBUS – DP) и скорость передачи. Скорость передачи должна совпадать с заданным в CFC значением и ограничивается в нашем примере ET200U, а также коммуникационным модулем CB1 SIMOVERT'a Master Drive (в нашем случае: 1,5 Мбод).

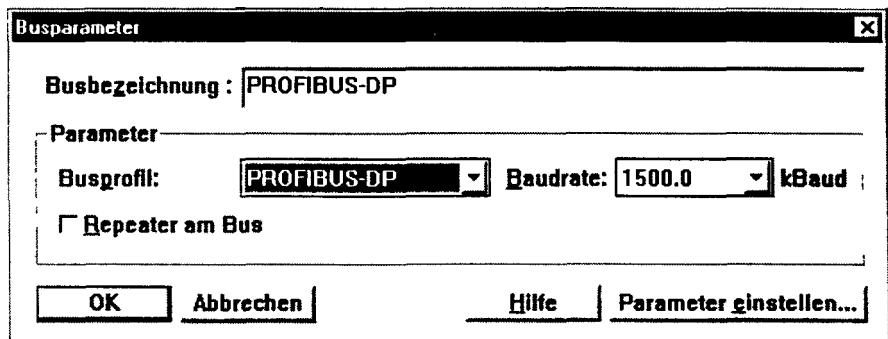


Рис 5.4-20. Диалоговое поле "Busparameter" ("Параметры шины").

3. На этом проектирование SS52 для нашего примера конфигурации заканчивается и может быть сохранено.

**Перенос
спроектированной
базы в память
SS52**

Следующим шагом для организации конфигурирования SS52 является перенос спроектированной базы в память субмодуля коммуникационного SS52. Для этого предоставлены два способа:

- Трансфер через второй интерфейс субмодуля (RS232), который находится на том же 9-полюсном Sub – D – разъеме, что и RS485.
 - Трансфер может быть проведен через любой интерфейс PC (COM1 или COM2), причем запись в память SS52 (Download) передается специальной программе-загрузчику по имени "SS52load".
 - Этот загрузчик требует файла форматом "2bf", вследствии чего маркированная "Hostsystem <2>" ("Система узла <2>") должна быть конвертирована в правильный формат посредством выбора пункта меню **Datei-Export > Binardatei...** (при этом процессе узлы системы (Hostsysteme) трактуются как раздельные). При этом файл конфигурирования SS52, подготовленный для загрузки в субмодуль, размещается в корневой директории программы COM PROFIBUS в поддиректории "\progdat".
- Трансфер через интерфейс PROFIBUS RS485 (поддерживается непосредственно COM PROFIBUS).
 - Трансфер по RS485 возможен, если PC оснащен специальной интерфейсной платой (например CP 5411).

5.4.8 Загрузка СОМ – базы данных на SS52.

**Необходимые
аппаратные
средства**

Необходимые аппаратные средства для загрузки СОМ – базы данных на SS52:

- Связь по RS232 между PC и SS52
 - На 9 – полюсный Sub – D – разъем помимо RS485 дополнительно встроена микросхема интерфейса RS232.

Дополнительную информацию

по Sub – D – штеккеру смотри документацию пользователя "SIMADYN D, Hardwarebeschreibung" ("SIMADYN D, Описание аппаратных средств").

- Так как в данномслучае речь идет о нестандартизированной распайке разъема, необходимо самостоятельно изготовить "перекрещенный" кабель (TxR на RxT) по прилагаемой схеме.

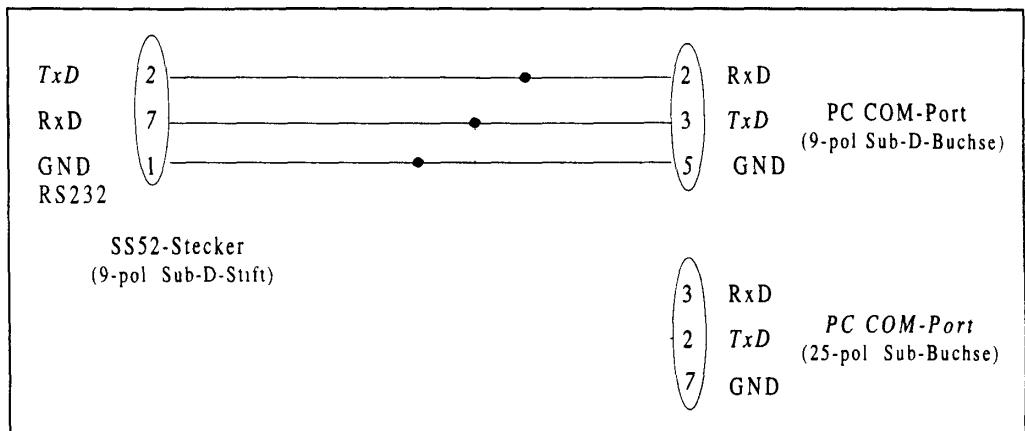


Рис 5.4-21. Схема кабеля RS232.

5.4.9 Работа с загрузчиком "SS52load".

SS52load

SS52load встроен в COM PROFIBUS (с версии 3.1).

В распоряжение пользователя предоставляются следующие функции:

- **Opnion-Comport** (опция-Comport): определение используемого Com-порта
- **Datei-Download** (файл-загрузка): выбор желаемого файла и подключение к загрузчику.

5.4.10 Поведение SS52 во время и после загрузки.

Общие положения

Для успешного проведения загрузки необходимо знать различные состояния SIMADYN D и, соответственно, коммуникационного субмодуля SS52 перед, во время и после процесса загрузки. Общее состояние системы индицируется зеленым и желтым светодиодами, которые находятся рядом с каждым местом установки субмодуля на CS7.

Эти светодиоды дают информацию исключительно о том, корректно ли работает SIMADYN D как отдельная, вне связи с другими станциями, система (в тексте – дословно: **закрытая система**) и, соответственно, где находятся возможные ошибки. При этом активность шины и связь с другими абонентами не оценивается.

- При подаче напряжения вначале в течение короткого времени горят оба светодиода (примерно полсекунды).
- Затем желтый светодиод постепенно гаснет, в то время как зеленый светодиод продолжает гореть на протяжении всего оставшегося времени инициализации (примерно пять секунд).
- По окончании фазы включения в работу индицируется рабочее состояние SS52¹.

¹ Подробнее см. стр.18.

SIMBOOK, раздел "Kommunikationsprojektierung D7-SYD".

Коммуникация с WinCC (SINEC H1).

Введение.

Данное руководство представляет в Ваше распоряжение простой пример проекта возможной связи Simadyn'a с WinCC по шине SINEC H1. Описаны все необходимые этапы проектирования (включая условия начала проектирования по аппаратному и программному обеспечению). Инструкции по работе с необходимым программным обеспечением здесь не приводятся, однако имеются ссылки на соответствующие руководства.

Условия начала проектирования.

Программное обеспечение.

Драйвер PC	SIMATIC NET TF1413 V4.01 для WIN95 V4.0
Simadyn D-PMC Kanal DLL для WinCC V4.0	Версия продукта 4.00.01.03
Пакеты (Tools)	ADRIMP (beinhaltet PROBI) для WinCC 4.0 V4.00.01.03 (включен в пакет Simadyn D-PMC Kanal DLL)
	Пакет проектирования SINEC NML для CSH11 V3.01

Примечание.

Simadyn D-PMC Kanal DLL может быть применен только для связи с WinCC V4.0. В приведенном здесь примере проектирования использовался Kanal-DLL с Simadyn D PMC Ethernet Layer 4 (WIN95/NT4.0, TF1413 V3.1). Инсталляция проводится посредством Setup, который находится на дискетах программных компонентов.

PMC Kanal DLL встраивается в структуру данных WinCC только тогда, когда инсталлирован соответствующий коммуникационный драйвер SIMATIC NET 1413. Инсталляция производится посредством Setup, находящегося на CD-ROM с версией пакета SIMATIC NET.

Аппаратное обеспечение.

Для PC, где проектируется WinCC:

- Кабель SIMATIC NET 727-1 для Industrial Ethernet!
- Сетевая карточка SIMATIC NET CP1413.

Примечание.

В PC должен быть инсталлирован CP1413 (muß sein!!!).

Литература

№	Наименование	Заказной номер	Поставщик
1	Драйвер коммуникации SIMADYN D-PMC для WinCC, руководство пользователя	UST-ID LIEF:DE129274202	SIEMEN A&D SH 7 Erlangen
2	SINEC NML-CP1413,CP1470 Пакет программирование для CP1413/CP1470 включая руководство	6GK1 740-0AB00-0BA0 (нем) 6GK1 740-0AB01-0EA0 (англ)	A&D AS EWK PLZ Karlsruhe A&D SE B3 KT Fuhrt
3	Пакет SIMATIC NET TF-1413 для TF-протокола	6GK1701-1TH00-3AA0	A&D AS LZN Nurnberg
Изм	Лист № <u>7</u> руководство пользователя на CD	6DD1987-0AA2	A&D LD R Erlangen
Руковод.			Лит.
Исполн.			Лист
Исполн.			Листов
Н.контр			1
Утв			
			Автоматизированные системы и комплексы
			

5	WinCC руководство пользователя	AV6 392-1XA03-0AA0 (нем) 6AV6 392-1XA03-0AB0 (англ) 6AV6 392-1XA03-0AC0 (фр)	A\$D S12 Nurnberg
---	--------------------------------	--	-------------------

SIMADYN D. Конфигурация аппаратной части.

Система	SIMADYN D	
Корзина	SR6-V	6 мест вставки, с вентилятором
Место вставки 1	PM6	CPU (с локальным интерфейсом для сервиса)
Место вставки 1.1	MS51	Карточка памяти 4Mbyte Flash
Место вставки 2	MM11	Буферная память
Место вставки 3	CSH11	Присоединение к SINEC H1

Переменные процессы.

Средствами CFC станция SIMADYN конфигурируется, параметрируется и для нее может быть создан любой тестовый план (тестовая программа). Конфигурация аппаратной части описана выше. В нашем случае процесс CFC-проектирования для станции SIMADYN не описывается детально. Обращайтесь за подробностями к руководству по проектированию.

Проектирование SIMADYN D.

CFC-план для связи с WinCC не является отдельным планом, однако его желательно делать таковым для наглядности. Для обмена данными (переменными процесса) между SIMADYN и WinCC требуются следующие программные модули:

- LI – модуль интерфейса LAN
- VM – модуль визуализации
- VI – модуль интерфейса
- VC – модуль-концентратор
- CI – модуль интерфейса
- SER02 – модуль коммуникации

Модуль связаны между собой следующим образом (описаны только используемые коннекторы):

Модуль LI

Имя коннектора	Значение	Пример
CTS	Имя используемого присоединения (CSH11)	D0300C
AT	Имя канала: ATC01. Уровень 4: #4 SDCOR1: имя NML-связи (смотри проектирование в NML)	'ATC01.#4SDCOR1'
AR	Имя канала: ARC01. Уровень 4: #4 SDCOR2: имя NML-связи (смотри проектирование в NML)	'ARC01.#4SDCOR2'
NA	Макс. число параллельных задач в WinCC	15
NC	Идентификатор WinCC	0
COM	Среда коммуникации H1	0
CCV	Связь с VM, коннектор CVP	<VM.CVP
CCF	Нет связей с модулем FM	16#0
CCB	Нет связей с модулем MM	16#0

Модуль VM

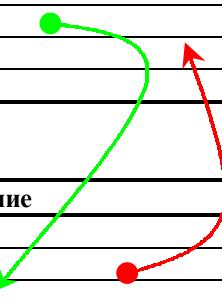
Имя коннектора	Значение	Пример
NA	Сумма зарезервированных для VM задач	40
NL	Число модулей LI	1
NV	Число модулей VI	1
MEM	По умолчанию	0
TGL	По умолчанию	0
CVP	Связь с LI.CCV, VI.CCV	>(LI.CCV, VI.CCV)

Модуль VI

Имя коннектора	Значение	Пример
CTS	Имя процессора	D01_P1
AT	Имя канала передачи на VC	'CMDVCH'
AR	Имя канала приема от VC	'ACKVCH'
CCV	Связь с VM, коннектор CVP	<VM.CVP

Модуль VC

Имя коннектора	Значение	Пример
CTS	Имя процессора	D01_P1
AT	Имя канала передачи на VI	'ACKVCH'
AR	Имя канала приема от VI	'CMDVCH'
NC	Число присоединенных модулей CI	1
CVP	Связь с CI, коннектор CCV	>CI.CCV



Модуль CI

Имя коннектора	Значение	Пример
CTS	Имя процессора	D01 P1
AT	Имя канала передачи на SER02	'CMDH'
AR	Имя канала приема от SER02	'ACKH'
ADT	Канал данных от SER02	1
CVP	Связь с VC, коннектор CVP	<VC.CVP

Модуль SER02

Имя коннектора	Значение	Пример
CTS	Имя процессора	D01 P1
AT	Имя канала передачи на CI	'ACKH'
AR	Имя канала приема от CI	'CMDH'
ADT	Имя канала передачи данных на CI	'DATH'
CLT	Длина канала передачи	116
CLR	Длина канала приема	524
TPD	Для В&В (обслуживания и наблюдения)	0
NL	Макс. число листов измеряемых величин	40
NV	Макс. число измеряемых величин (коннекторов)	500

Примечание. Дополнительно должны быть спроектированы функциональные модули @CMM для буферной памяти и модуль инициализации связи @CSH11.

Модуль @CSH11

Имя коннектора	Значение	Пример
CTS	Имя примененной карточки (CSH11)	'D0300C'
MAA	Адрес Simadyn D на шине Industrial Ethernet	080006010002
CDV	Реорганизация памяти (1)	0

Модуль @CMM

Имя коннектора	Значение	Пример
CTS	Имя модуля буферной памяти	'D0200A'
CDV	Реорганизация памяти (1)	0

Проектирование WinCC.

Для примера проектирования достаточно простого проекта WinCC с объединенными полями ввода/вывода.

Описание проектирования WinCC здесь не приводится. Подробное описание работы с WinCC Вы можете прочитать в соответствующем руководстве по проектированию. Для создания первого проекта на WinCC достаточно руководства SIMATIC WinCC Getting Started.

Организация битовых сообщений (аварийных сообщений).

Проект Simadyn D.

Для организации битовых сообщений с WinCC не требуется специального проектирования для отображения значений переменных процесса. Выбор того, какому биту переменной соответствует данное сообщение, производится исключительно в WinCC. Предписания по проектированию для выдачи переменных процесса остаются в силе.

Проект WinCC.

Дополнительно к проектированию переменных процесса необходимо провести проектирование ALARM-Logging. Описание проектирования WinCC здесь не приводится. Подробное описание работы с WinCC Вы можете прочитать в соответствующем руководстве по проектированию. Для создания первого проекта на WinCC достаточно руководства SIMATIC WinCC Getting Started.

Сообщения SIMADYN D (не аварийные сообщения).

Проект Simadyn D.

Для выдачи сообщений из Simadyn'a на WinCC дополнительно к проектированию передачи данных процесса необходим программный модуль MM (специально для WinCC):

- MM – менеджер сообщений.

Модуль связан с другими модулями следующим образом (описаны только используемые коннекторы):

Модуль MM

Имя коннектора	Значение	Пример
CTS	Имя процессора	D01_P1
AR	Имя канала (идентично заданному на коннекторе AT модуля MSI)	EMPFKANA
NZ	Число циклов на передачу	5
NL	Число присоединенных модулей LI	1
MEM	Цикл диагностики	0
TGL	Цикл диагностики	0
CVP	Связь с LI, коннектор CCV	>LI.CCV

Примечание. Дополнительно необходимо спроектировать центральный модуль системы сообщений @MSI, модуль передачи сообщений MSI и модуль сообщений MERFO.

Модуль @MSI

Имя коннектора	Значение	Пример
CMS	Имя системы сообщений	MYMELD
CMT	Текст сообщения (здесь – не задан)	" "
NOM	Число сообщений, которые можно запомнить	200
SAV	Буфер сообщений в буферной RAM	0
RP	Префикс для коммуникационной ошибки	0
MUN	Разблокировка передачи сообщений	1

Модуль MSI

Имя коннектора	Значение	Пример
CMS	Имя системы сообщений	MYMELD
CTS	Имя модуля, куда присоединена система сообщений	D01_P1
AT	Параметр адреса	EMPFKANA
RP	Префикс сообщений, переполнивших буфер сообщений	0
SNV	Выдача номера сообщения	1
STM	Выдача текста сообщения	0
STC	Выдача текста сообщения фиксированной длины	1
SSF	Выдача формата	1
EN	Разблокировка	1
MUN	Разблокировка передачи сообщений	1

Модуль MERFO

Имя коннектора	Значение	Пример
CMS	Имя системы сообщений	MYMELD
MT	Тип сообщения	1
RP	Префикс	0
RS1	Суффикс приходящего сообщения	10001
RS2	Суффикс выходящего сообщения	00005
EN	Разблокировка сообщения	1
IS1	Триггер сообщения	16#0
SM	Запоминание сообщения	0

Проект WinCC.

Дополнительно к проектированию переменных процесса необходимо провести проектирование ALARM – Logging. Описание проектирования WinCC здесь не приводится. Подробное описание работы с WinCC Вы можете прочитать в соответствующем руководстве по проектированию. Для создания первого проекта на WinCC достаточно руководства SIMATIC WinCC Getting Started. Соответствие номеров сообщений, заданных в Simadyn на коннекторах RS* модуля сообщений MERFO, номерам сообщений, объявленных в WinCC, предусматривается только посредством "PMC-message no", которое создается для номеров сообщений в листе сигналов.

Создание адресной книги в редакторе CFC.

Для создания листа сигналов для WinCC необходима ADRIMP символьная информация процессоров Simadyn. Simadyn создает для каждого CPU ASCII – файл, где содержится эта информация. Имя файла состоит из имени корзины и имени CPU, разделенных знаком "_". В качестве расширения используется .ADR.

Адресная книга создается посредством выбора желаемого плана проекта и выбора пункта меню "Extras – Einstellen – Uebersetzen". Затем необходимо отметить опцию "Создать адресную книгу - Adreßbuch erzeugen" и закрыть пункт меню посредством ОК. Далее вызвать пункты меню "Plan – Uebersetzen", т.е. скомпилировать план. Адресная книга будет создана только при компиляции. Созданная для данного плана адресная книга находится под пунктом меню "Extras – Protokolle".

Ограничения. Необходимой предпосылкой для создания адресной книги пока является соблюдение старого синтаксиса STRUG G. В связи с этим следует обращать внимание на следующие ограничения, накладываемые на синтаксис в CFC:

- В задании имени используются только заглавные (большие) буквы
- Имя плана: макс. 6 символов, первый символ должен быть буквой
- Имя модуля: макс. 6 символов, первый символ должен быть буквой
- Имя коннектора: макс. 3 символа, первый символ должен быть буквой
- Имя \$-переменной: макс. 6 символов, первый символ должен быть буквой
- Область значений: 0,00001 – 99999

NML – проектирование для CSH11.

Конфигурация шины для модуля CSH11 (с встроенным CP1470) проводится пакетом программирования NML. Пользование пакетом NML здесь также не описывается, отсылаем Вас к соответствующему руководству по NML.

Следующий ниже проект проектирования достаточен для построения связи WinCC с Simadyn D через карточку CSH11 по протоколу SINEC H1 (Industrial Ethernet – уровень 4).

Пример проектирования. SIEMENS AG SINEC NML V3.01 документация по коммуникациям 19.02.1998.

- Тип узла: CP147x
- Имя узла: roland
- Страница: 1/1

Основная характеристика узла

- Тип интерфейса: CP147x
- Имя интерфейса: CP147x
- Профиль интерфейса: CSH11_E4_2000
- Адрес на шине: 0800006010002 (SIMADYN)
- Транспортных соединений: 6
- Связей приложений: 26
- Длина базы данных: 5888Byte
- Число таблиц FVT's: 1

Определение коммуникаций

1. Транспортное соединение SDCOR1

- Локальный TSAT – ID: AG_WINCC
- Удаленный TSAT – ID: WINCC_AG
- Удаленный адрес на шине: 0800006010001 (PC)
- Построение связи: aktiv -> dyn.
- Тип связи: E4 – Verbind.
- Физическая связь: Bus
- Имя профиля: e4_handshake
- Нет подключенных приложений пользователя

2. Транспортное соединение SDCOR2

- Локальный TSAT – ID: WINCC_AG
- Удаленный TSAT – ID: AG_WINCC
- Удаленный адрес на шине: 0800006010001 (PC)
- Построение связи: passiv <- dyn.
- Тип связи: E4 – Verbind.
- Физическая связь: Bus
- Имя профиля: e4_handshake
- Нет подключенных приложений пользователя

При помощи утилиты **Загрузка (Transferieren)** можно провести загрузку описания коммуникации, созданной в приведенном выше проекте коммуникации, в модуль CSH11.

Инструмент импорта адресного листа ADRIMP.

Инструмент импорта адресного листа ADRIMP необходим для того, чтобы WinCC могла интерпретировать имена указателей Simadyn D. Инструмент импорта адресного листа ADRIMP открывает возможность встраивания в банк данных WinCC текстового листа адресов (TALI). Подробное описание Вы найдете в руководстве пользователя на коммуникационный драйвер SIMADYN D --PMC для WinCC. У нас этот файл лежит на сервере (Server\Архив\Фирмы\Siemens\Simadyn D\SDPMSdeu.doc).

Необходимые условия.

Должен существовать файл определения переменных и к нему должна быть создана адресная книга Simadyn. Файл определения переменных и адресная книга должны лежать в соответствующих каталогах. Каталог генерации может быть другим, однако для большей наглядности можно проводить генерацию в этом же каталоге. (?)

Создание файла определения переменных.

Файл определения переменных является текстовым файлом и должен быть создан пользователем. Файл определения переменных состоит из двух определенных синтаксисом заголовков (Kopfzeilen) (1. и 2.), вытекающих из согласования символьных имен (переменных WinCC) именам коннекторов Simadyn D. Символьные имена выбираются произвольно, однако для наглядности должны быть такими, чтобы их можно было использовать в текстовых полях WinCC.

Выдержки из файла определения переменных.

Например: winccvar.txt

- 1). [VDM:wincc] – имя партнера по коммуникации.
- 2). [PN:A000_1,C:\wincc\vardatei]
- 3). MOTOR_EIN, ANBIND.CI.CCV
MOTOR_AUS, ANBIND.CI.YTS

Создание и импорт нового листа сигналов.

Необходимые условия.

Для создания и импортирования листа сигналов в проекте WinCC должен быть инсталлирован драйвер SIMADYN D – PMC.

- Вызвать проект WinCC
- Кликнуть по менеджеру переменных (Variablenhaushalt).
- Отметить пункт меню "Добавить новый драйвер"
- Выбрать SIMADYN D PMC Ethernet.chn

Если перед стартом ADRIMP не был запущен проект WinCC, при импортировании листа сигналов будет использован последний обрабатываемый проект.

Процесс.

- Вызвать ADRIMP
- Выбрать пункт меню "File" (Файл)
- Выбрать пункт меню "Probi"
- Найти файл определения переменных (например: winvar.txt)
- Назначить каталог, куда будет сгенерирован файл

- Создать лист сигналов (например: wincc.txt)
- Завершить Probi

Примечание. Лист сигналов автоматически импортируется ADRIMP в соответствующую базу данных менеджера данных WinCC.

Импорт имеющегося листа сигналов.

- Стартовать желаемый проект WinCC
- Вызвать ADRIMP
- Выбрать пункт меню "File" (Файл)
- Выбрать пункт меню "Open"
- Выбрать лист сигналов (например: wincc.txt)
- Завершить ADRIMP

Примечание. Лист сигналов автоматически импортируется ADRIMP в соответствующую базу данных менеджера данных WinCC.

Проверка созданной в WinCC базы данных.

Проверка импортированных данных, их имен, форматов данных и имен указателей Simadyn D:

- Вызвать проект WinCC
- Выбрать менеджер данных
- Кликнуть по "логическая связь" (с соответствующим VDM – именем)
- Выбрать SIMADYN D PMC ETHERNET
- Выбрать SD- PMC (CP1413-1)
- Отметить логические имена связей

Будут подсвечены логические имена и имена указателей Simadyn D, определенные в файле переменных. Дополнительно будет показан формат данных. Только к этим переменным WinCC будет иметь доступ.

Организация связи SIMADYN D - WinCC.

Шина связи.

Физически связь между SIMADYN D и WinCC осуществляется посредством кабеля SIMATIC NET 727-1 для Industrial Ethernet.

Активизация WinCC.

Для создания связи между WinCC и Simadyn необходимо установить соответствие между импортированными в банк данных WinCC данными и полями входов\выходов их графического проектирования (представления). Это производится посредством выбора соответствующих полей в Graphics Designer и открывания диалогов конфигурирования. Каждому полю поставлена в соответствие одна импортированная переменная. После проведения согласования в главном меню отмечается пункт "Файл" и данные запоминаются. Перед стартом исполняемой части должны быть установлены свойства связи.

B Control Center.

- Кликнуть "Менеджер переменных" (Variablenhaushalt)
- Кликнуть "Свойства"
- Кликнуть "свойства канала Unit" (Eigenschaften Kanal Unit)
- Кликнуть "Связь" (Verbindung)

Внести Ethernet-адрес PC (см. проект NML).

Recv Funktion: внести собственный TSAP-ID (см. проект NML).

Send Funktion: внести собственный TSAP-ID (см. проект NML).

- Подтвердить посредством OK

- Кликнуть "Активировать"

Теперь WinCC готов к обмену данными с Simadyn.

Активизация Simadyn.

Включить спроектированную корзину. В процессе работы установится связь между Simadyn D и WinCC. Теперь Simadyn D и WinCC будут циклически обмениваться данными.

Simadyn D-OS Ingineerimg.

Аппликация "Simadyn D-OS Ingineerimg" предназначена для передачи информации о переменных, предназначенных для управления и визуализации, на станцию оператора (WinCC), где эта информация будет использована в дальнейшем. Программа передачи данных "Simadyn D-OS Ingineerimg" входит в пакет программного обеспечения "D7-SYS". Описание работы с "Simadyn D-OS Ingineerimg" приводится в C:\SIEMENS\STEP7\S7BIN\p7sdmapa.hlp.

06.06.01.

Аппликация "Simadyn D-OS Ingineerimg" работала с WinCC V3.0 и будет работать с WinCC с более высокими версиями, чем V5.x. Сегодня для привязки данных CFC-Simadyn к WinCC надо пользоваться инструментом ADRIMP. ADRIMP (включая PROBI) поставляется вместе с дискетой SD-PMC Kanal DLL.

Коротко работу с ADRIMP можно описать следующим образом:

- После создания CFC-пакета, перед началом компиляции необходимо установить опцию компиляции "Создать адресную книгу" (Extras => Einstellungen => Adreßbuch Erzeugen). После проведения компиляции будет автоматически сгенерирована адресная книга. Файл адресной книги находится

Разные заметки по связи Simadyn – WinCC.

Die PMC Kanal-DLL kann nur dann in den Variablenhaushalt von WinCC eingefügt werden, wenn der zugehörige SINEC Kommunikations-treiber installiert ist!

PMC Kanal –DLL будет поддерживать обмен данными с WinCC только тогда, когда инсталлирован соответствующий коммуникационный драйвер SINEC!

PMC Kanal –DLL должен применяться со следующими операционными системами и драйверами:

SIMADYN D PMC Ethernet (Win95, **TF1413 V4.01** или NT4.0, **TF1413 V5.0**).

