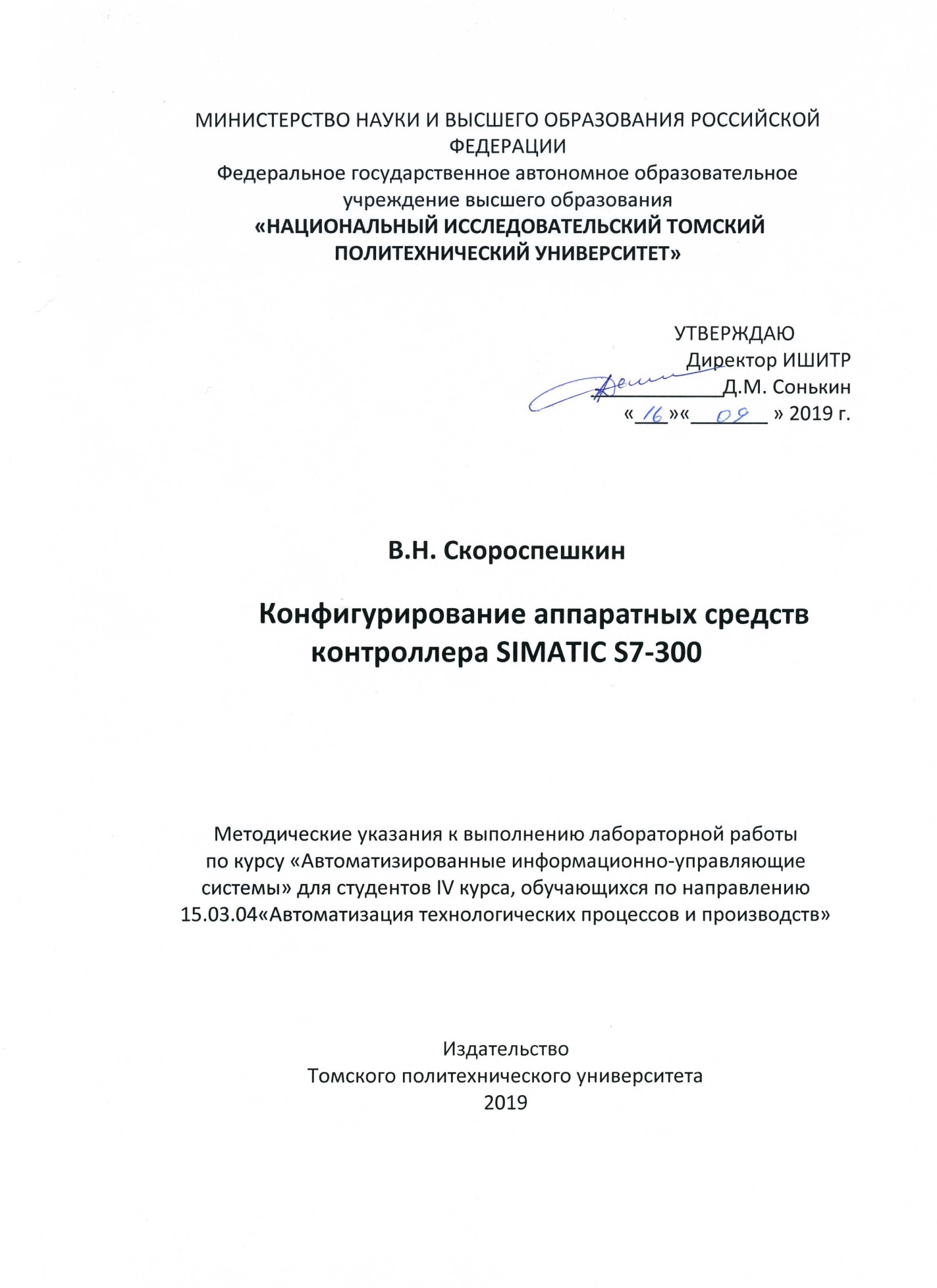
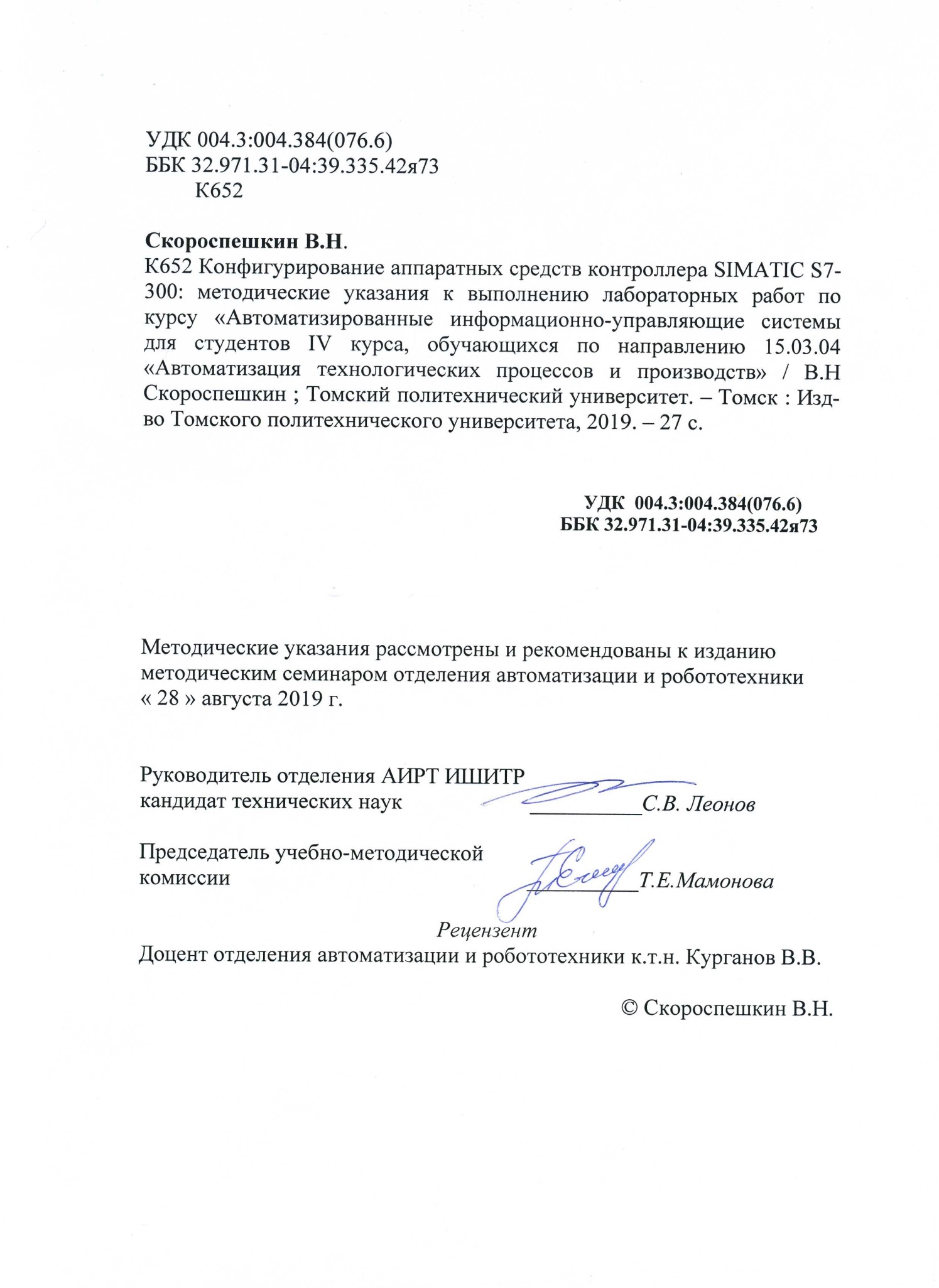
****

****

**Цель работы:**

Изучение основ технологического программирования контроллера в пакете STEP 7.

**1. Структурная схема лабораторного стенда**

SIMATIC S7-300 – это модульный программируемый контроллер для решения задач автоматизации. Несколько типов центральных процессоров различной производительности и широкий спектр модулей различного назначения с множеством встроенных функций позволяют выполнять максимальную адаптацию оборудования к требованиям решаемой задачи.

При модернизации и развитии производства контроллер может быть легко дополнен набором необходимых модулей. На рис. 1 приведена структурная схема лабораторного стенда, позволяющая исследовать контроллер SIMATIC S7- 300.

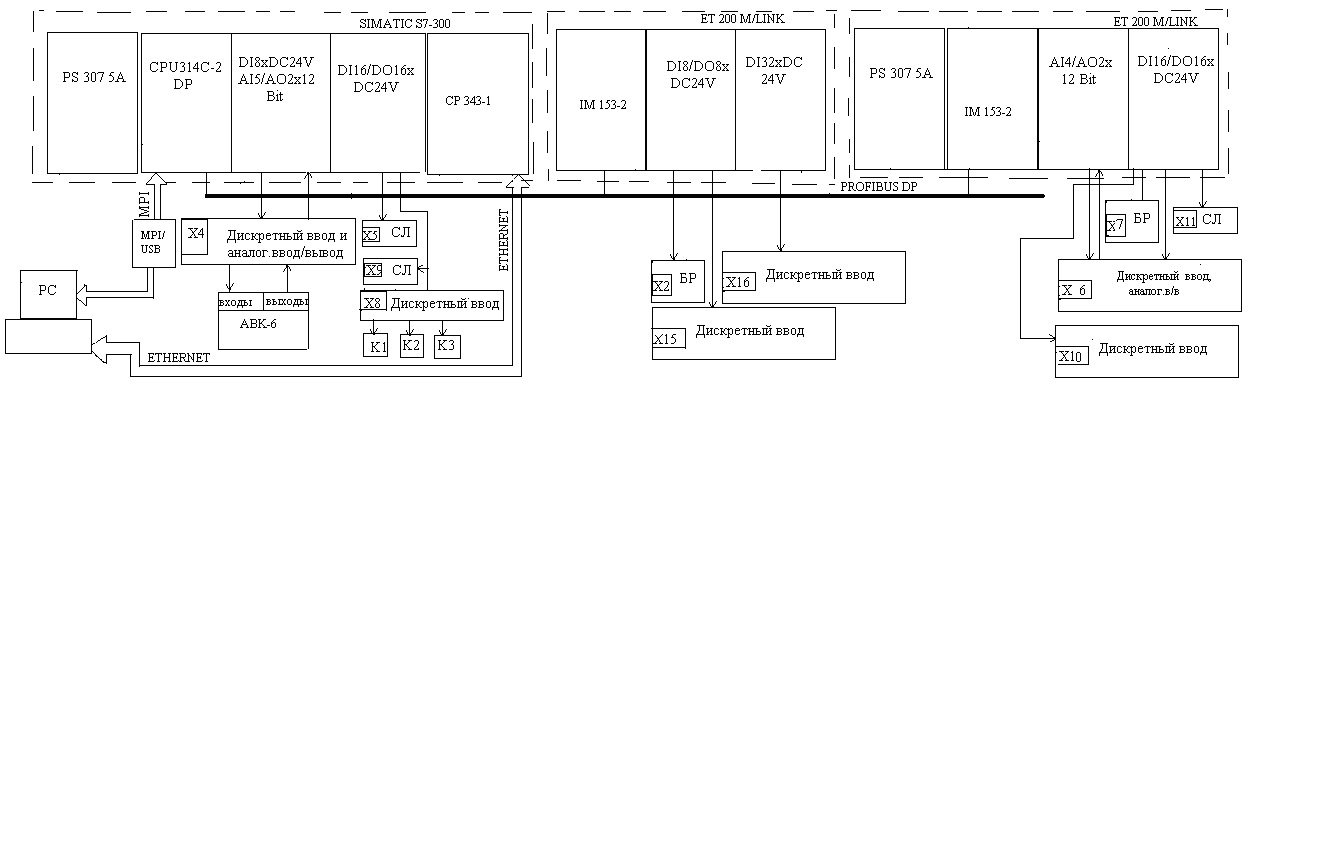
В данный лабораторный комплекс входят модули контроллера- блок питания PS 307 5A (307-1EA00-0AA0), центральный процессор CРU 314С - 2 DP (314-6CG03-0AB0) c встроенными модулями дискретного ввода DI8xDC24V и аналогового ввода/вывода AI5/AO2x12Bit, коммуникационный модуль СP 343-1, предназначенный для организации последовательной передачи данных по Ethernet интерфейсу.

В состав комплекса входят также станции распределенного ввода ET200M/LINK (IM 153-2) с заказным номером 153-2BA02-0XB0, модули ввода/вывода дискретного сигнала DI 32xDC24V (321-1BL00-0AA0) и DI8/DO8xDC24V (323-1BH01-0AA0), модуль аналогового ввода/вывода AI4/AO2x12Bit (334-0KE00-0AB0), модуль дискретного ввода/вывода DI16/DO16xDC24V (323-1BL00-0AA0),а также 2 блока реле (X2, X7).

Для исследования свойств систем регулирования в состав комплекса вклечен АВК аналогового – вычислительный комплекс АВК-6 . Каналы модуля ввода аналогового сигнала поступают на клеммную колодку Х4, где осуществляется ввод сигнала с АВК. Каналы модуля ввода дискретного сигнала поступают на колодки Х15, Х16, X8 и X10.

На колодке Х8 осуществляется ввод дискретного сигнала при помощи кнопок. Каналы модуля вывода дискретного сигнала поступают на блоки реле Х2 и Х7, где осуществляется вывод дискретного сигнала при помощи сигнального табло. Тип клеммных соединений – «сухой контакт».

Назначение данного лабораторного стенда – ввод аналогового сигнала, ввод/вывод дискретного сигнала, то есть реализация программно-логического управления САУ с дискретными сигналами и контроль за САУ с аналоговыми сигналами.



*Рис. 1. Структурная схема лабораторного стенда*

**2. Описание программной среды STEP 7**

STEP 7 - это базовый пакет программ, включающий в свой состав весь спектр инструментальных средств, необходимых для программирования и эксплуатации систем управления, построенных на основе программируемых контроллеров SIMATIC S7/C7, а также систем компьютерного управления SIMATIC WinAC.

STEP 7 поставляется в трех вариантах:

- STEP 7 Lite - облегченная версия, используемая для программирования только SIMATIC S7-300 и SIMATIC C7;

- STEP 7 - полная версия для приложений, связанных с применением всех систем автоматизации SIMATIC;

- STEP 7 Professional - это пакет программ. В состав пакета входят STEP 7, S7-SCL, S7-GRAPH и S7-PLCSIM.

Отличительной особенностью пакета STEP 7 является возможность разработки комплексных проектов автоматизации, базирующихся на использовании множества программируемых контроллеров, промышленных компьютеров, устройств и систем человеко-машинного интерфейса, устройств распределенного ввода-вывода, сетевых структур промышленной связи. Ограничения на разработку таких проектов накладываются только функциональными возможностями программаторов или компьютеров, на которых установлен на STEP 7.

Инструментальные средства STEP 7 позволяют выполнять:

* конфигурирование и определение параметров настройки аппаратуры;
* конфигурирование систем промышленной связи и настройку параметров передачи данных;
* программирование, тестирование, отладку и запуск программ отдельных систем автоматизации, а также их локальное или дистанционное обслуживание;
* документирование и архивирование данных проекта;
* функции оперативного управления и диагностирования аппаратуры.

STEP 7 входит в комплект поставки программаторов SIMATIC Field PG и SIMATIC Power PG. Он может поставляться в виде самостоятельного пакета программ для персональных компьютеров, работающих под управлением операционных систем Windows 95/ 98/ NT/ ME/ 2000PROF/ XP PROF. Для возможности подключения программируемых контроллеров компьютер должен быть оснащен MPI картой или PC/MPI адаптером и соединительным кабелем.

STEP 7 обеспечивает параллельное выполнение работ по одному проекту несколькими разработчиками. Единственным ограничением при этом является невозможность одновременной записи одних и тех же данных несколькими разработчиками.

**2.1. Состав пакета STEP 7**

STEP 7 содержит полный спектр инструментальных средств, необходимых для выполнения всех этапов разработки проекта, а также последующей эксплуатации системы управления:

* SIMATIC Manager – ключевая программа STEP 7, позволяющая выполнять управление всеми составными частями проекта, осуществлять быстрый поиск необходимых компонентов, производить запуск необходимых инструментальных средств;
* Symbol Editor – программа определения имен переменных, типов данных, ввода комментариев к переменным;
* Hardware Configuration – программа конфигурирования используемой в проекте аппаратуры;
* Communication – программа конфигурирования систем промышленной связи, использующих для обмена данными MPI интерфейс, сети PROFIBUS, PROFINET или Industrial Ethernet. Сеансы связи могут осуществляться циклически или запускаться по временным или аппаратным прерываниям;
* Информационные функции – для быстрого доступа к данным центрального процессора и управления режимами выполнения программы пользователя в ходе ее отладки.

Для разработки программ пользователя STEP 7 позволяет использовать следующие способы их представления:

* список инструкций (Statement List – STL). Программы, написанные на STL, занимают минимальный объем в памяти программ контроллеров и обладают наиболее высоким быстродействием.
* диаграммы лестничной логики (Ladder Diagram – LAD). В отечественной литературе этот язык известен как язык релейно-контактных схем – РКС;
* язык функциональных блоков (Function Block Control Diagram – FBD). Язык, позволяющий выполнять разработку программы по аналогии с разработкой функциональной схемы устройства управления, создаваемого на основе интегральных логических элементов, счетчиков, таймеров и т.д.

Для решения специальных задач управления могут быть использованы дополнительные технологически ориентированные языки программирования, а также инструментальные средства, позволяющие конвертировать программы STEP5 и TISOFT в программы STEP 7.

**2.2. Типы блоков STEP 7**

STEP 7 объединяет все файлы программ пользователя и все файлы данных в блоки. В пределах одного блока могут быть использованы другие блоки. Механизм их вызова напоминает вызов подпрограмм. Это позволяет улучшать структуру программы пользователя, повышать их наглядность, обеспечить удобство их модификации, перенос готовых блоков из одной программы в другую.

В составе программ STEP 7 могут быть использованы блоки следующих типов

* Организационные блоки (ОВ), которые осуществляют управление ходом выполнения программы. В зависимости от способа запуска (циклическое выполнение, запуск по временному прерыванию, запуск по событию и т.д.) организационные блоки разделяются на классы, имеющие различные уровни приоритета. Организационные блоки с более высокими уровнями приоритета способны прерывать выполнение блоков с более низкими приоритетными уровнями. Предусмотрена возможность детального описания события, вызывающего запуск организационного блока. Эта информация может быть использована в программе пользователя.
* Функциональные блоки (FB) содержат отдельные части программы пользователя. Выполнение функциональных блоков сопровождается обработкой различных данных. Эти данные, внутренние переменные и результаты обработки загружаются в выделенный для этой цели блок данных IDB(Instancedata Block). Управление данными, хранящимися в IDB, осуществляет операционная система программируемого контроллера.
* Для каждого функционального (FB) и системного функционального (SFB) блока операционная система контроллера создает служебный блок данных IDB. IDB генерируются автоматически после компиляции FB и SFB. Доступ к данным, хранящимся в IDB, может быть осуществлен из программы пользователя или из системы человеко-машинного интерфейса.
* Функции (FC) – блоки, которые содержат программы вычисления. Каждая функция формирует фиксированную выходную величину на основе получаемых входных данных. К моменту вызова функции все ее входные данные должны быть определены. Такой механизм позволяет использовать функции без блоков данных.
* Блоки данных (DB) предназначены для хранения данных пользователя. В отличие от данных, хранящихся в IDB и используемых одним блоком FB или SFB, глобальные данные, хранящиеся в DB, могут использоваться любым из программных модулей. В DB могут храниться данные, имеющие элементарный или структурный тип. Примерами данных элементарного типа могут служить данные логического (BOOL), целого (INTEGER), действительного (REAL) или других типов. Данные структурного типа формируются из данных элементарного типа. Для обращения к данным, записанным в DB, может использоваться символьная адресация.
* Системные функциональные блоки (SFB) – это функциональные блоки, встроенные в операционную систему блоки, встроенные в операционную систему центрального процессора (например, SEND/ RECEIVE). Эти блоки не занимают места в памяти программ контроллера, но требуют использования IDB.
* системные функции (SFC) – это функции, встроенные в операционную систему контроллера. Например, функции таймеров, счетчиков, передачи блоков данных и т.д.
* системные блоки данных (SDB) – это блоки для хранения данных операционной системы центрального процессора. К этим данным относятся параметры настройки системы и отдельных модулей (аппаратных модулей).

**3. Инструментальные средства STEP 7**

**3.1 SIMATIC Manager**

SIMATIC Manager обеспечивает управление всеми данными проекта независимо от типа системы автоматизации (SIMATIC S7/C7 или WinAC), в которой они используются. Для всех систем автоматизации используется общий набор инструментальных средств. Инструментальные средства, необходимые для обработки тех или иных данных, запускаются из SIMATIC Manager автоматически.

**3.2 Symbol Editor**

Редактор позволяет выполнять все необходимые операции по обслуживанию глобальных переменных (редактор не используется для обслуживания локальных формальных параметров, хранящихся в IDB). При редактировании глобальных переменных могут использоваться следующие функции:

- определение символьных обозначений и ввод комментариев для обрабатываемых входных и выходных сигналов, битов памяти и блоков;

- сортировка данных;

- импорт/экспорт данных их других программ Windows.

Запись символьной переменной обеспечивает доступ к соответствующим данным со стороны всех приложений. Изменение символа одного из параметров автоматически распознается всеми инструментальными средствами.

**3.3. Hardware Configuration**

Конфигуратор используется для выбора и настройки всей аппаратуры, используемой в проекте. Он поддерживает выполнение следующих функций:

- конфигурирование систем автоматизации методом выбора необходимых компоентов из электронного каталога и их объединения в единую систему;

- конфигурирование систем распределенного ввода-вывода. Производится теми же способами, что и системы локального ввода-вывода контроллера;

- настройка параметров центральных процессоров. С помощью меню могут корректироваться все необходимые атрибуты (стартовые характеристики, проверка времени сканирования программы и т.д.). Поддерживается обслуживание мультипроцессорных систем. Вводимые данные сохраняются в системных блоках данных центрального процессора;

- настройка параметров модулей. С помощью экранных форм могут быть определены все параметры настройки используемых аппаратных модулей. Аналогичные установки с помощью DIP переключателей становятся ненужными. Последующая замена модуля не требует повторной настройки его параметров;

- настройка параметров функциональных модулей (FM) и коммуникационных процессоров (CP). Производится с помощью специальных экранных форм, которые включены в комплект поставки соответствующих модулей.

Контролируя состав используемых модулей, система предотвращает ввод некорректных данных конфигурации. Она не позволяет размещать модули контроллера в слотах, отведенных для других целей, не позволяет выполнять настройки модулей, при которых они не могут работать и т.д. В силу могут вступить только корректно введенные параметры конфигурации системы управления.

**3.4. Communication Configuration**

Начиная с версии 5.0, в комплект поставки STEP 7 входят пакеты NCM S7 для Industrial Ethernet и NCM S7 для PROFIBUS.

Эти пакеты существенно расширяют функции конфигурирования систем промышленной связи.

Конфигуратор систем промышленной связи позволяет:

* Конфигурировать и отображать коммуникационные соединения.

Конфигурировать циклический обмен данными по MPI интерфейсу: определять партнеров по связи, назначать источники и приемники информации. Формирование системных блоков данных и управление передачей данных по MPI интерфейсу осуществляется автоматически.

* Выполнять описание обмена данными по прерываниям: определять партнеров по связи, выбирать из встроенной библиотеки коммуникационные блоки (CFB), настраивать выбранные коммуникационные блоки на языках LAD/STL/FBD.

**4. Системная диагностика**

Системная диагностика позволяет определить текущее состояние программируемого контроллера, а также выявлять ошибки в модулях и их внешних цепях. Подробная информация о дефекте может быть вызвана двойным щелчком кнопки мыши в момент позиционирования курсора на изображении модуля.

Объем и вид этой информации определяются типом модуля:

- отображение основной информации о модуле (заказной номер, версия, обозначение) и его состоянии;

- отображение информации об отказах модулей ввода-вывода систем локального и распределенного ввода-вывода;

- отображение сообщений из диагностического буфера.

Для центральных процессоров дополнительно может быть

проанализирована информация:

- о возникновении отказов в процессе выполнения программы;

- о продолжительности цикла выполнения программы;

- о доступном объеме используемой и свободной памяти;

- о доступных и используемых ресурсах MPI связи;

- а также служебная информация (допустимое количество входов и выходов, флагов, счетчиков, таймеров и блоков).

**5. Языки программирования**

Для программирования систем автоматизации SIMATIC S7/C7/WinAC может быть использовано три языка: STL, LAD и FBD.

Язык STL (Statement List) позволяет создавать наиболее компактные программы, обладающие наиболее высоким быстродействием. Язык поддерживает выполнение следующих функций:

- Поиск. Любая точка программы может быть быстро найдена по указанию символьного имени, оператора и т.д.

- Возможность ввода данных в пошаговом или свободном текстовом режиме. Программа может вводиться с проверкой синтаксиса каждой строки или набираться в текстовом редакторе с последующим преобразованием.

Классические языки программирования контроллеров LAD (Ladder Diagram - LAD) и FBD (Function Block Diagram) позволяют создавать программы, отвечающие требованиям DIN EN 6.1131-3.

Редакторы стандартных языков LAD и FBD обеспечивают полную графическую поддержку программирования со следующими характеристиками:

- простое и интуитивное использование, наглядный интерфейс, использование стандартные механизмы работы с Windows;

- библиотеки заранее подготовленных сложных функций (например, ПИД регулирования) и разработанных пользователем решений.

**6. Система команд**

STEP 7 оснащен исчерпывающим набором инструкций, позволяющим легко и просто решать любые задачи автоматического управления. Этот набор инструкций включает в свой состав:

- логические операции (включая обработку фронтов);

- операции со словами;

- операции с таймерами и счетчиками;

- операции сравнения;

- операции преобразования типов данных;

- операции сдвига и вращения;

- математические функции (включая тригонометрические и экспоненциальные);

- функции управления ходом выполнения программы.

Для облегчения программирования может быть использован целый ряд дополнительных сервисных функций:

- установка точек прерывания;

- управление состоянием входов и выходов;

- переключение режимов работы;

- отображение перекрестных ссылок;

- поддержка мультипроцессорных конфигураций.

Функции состояний:

- загрузка и тестирование блоков непосредственно в редакторе;

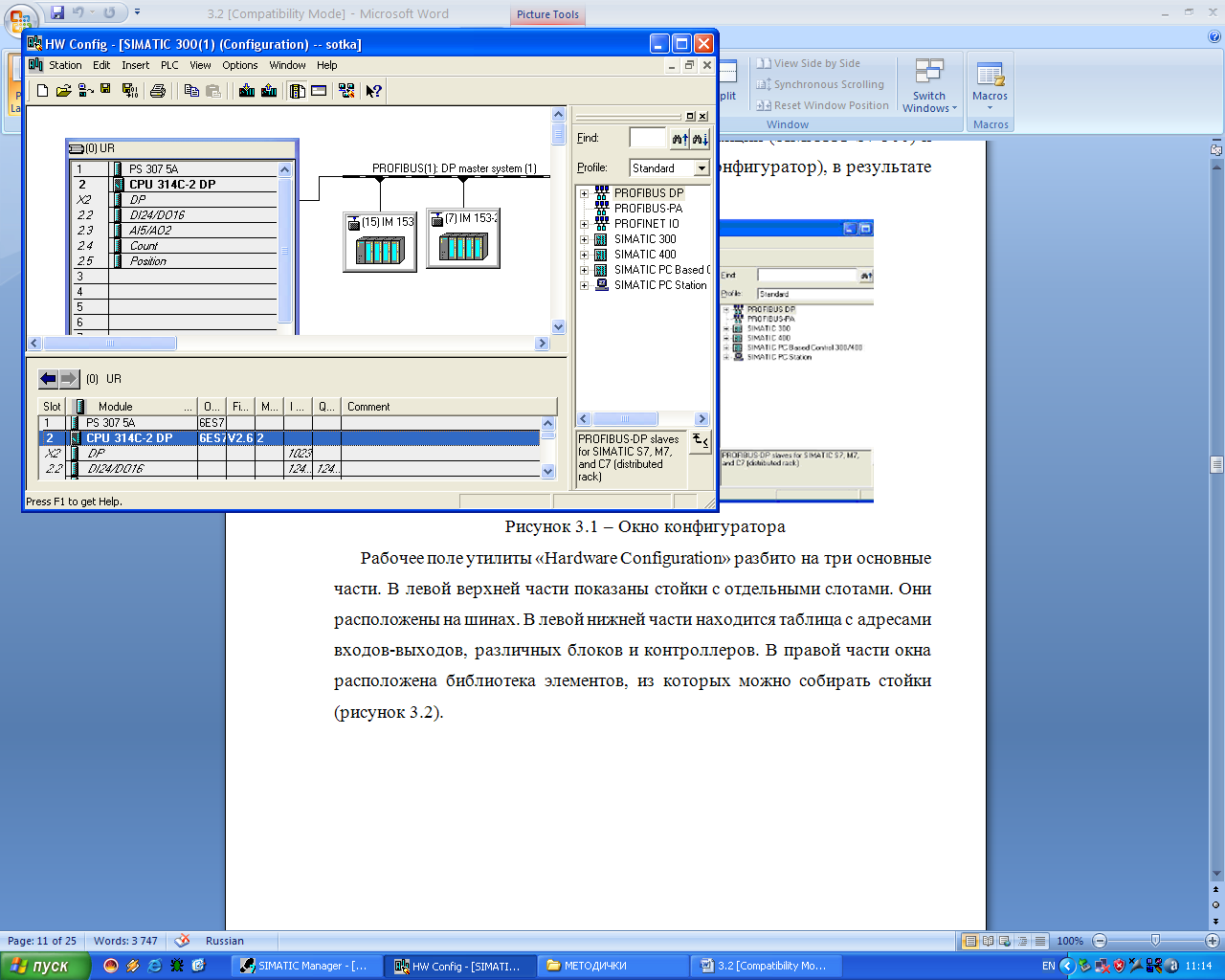
- одновременный контроль состояний нескольких блоков;

- функции поиска.

Доступна интерактивная помощь по всем функциональным блокам (F1).

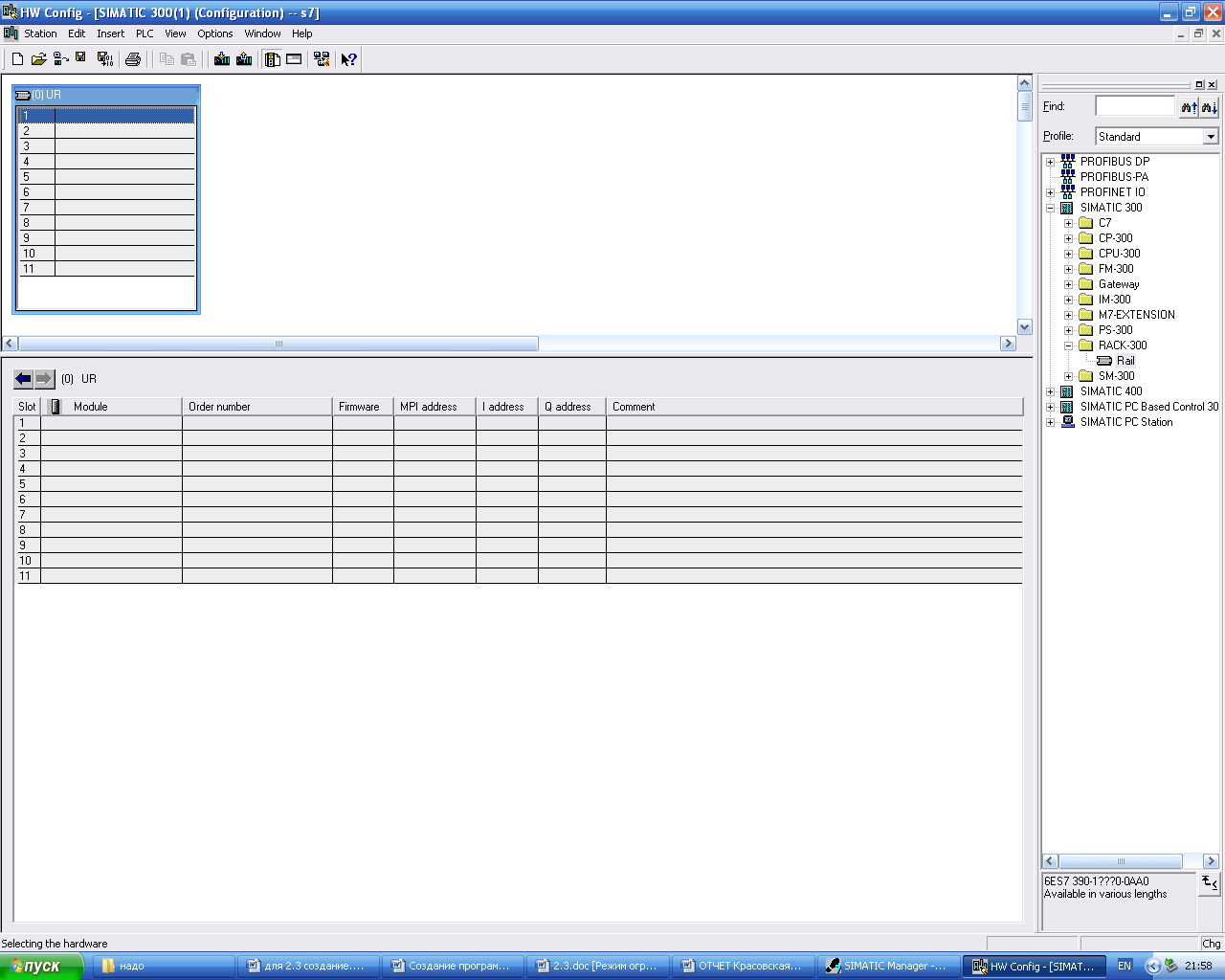
**7. Конфигурирование аппаратных средств**

Конфигурирование аппаратных средств проекта осуществляется посредством утилиты «Hardware Configuration» (программа конфигурирования аппаратных средств). Чтобы запустить указанную программу, необходимо перейти на уровень станций (SIMATIC S7-300) и двойным щелчком нажать кнопку «Hardware» (Конфигуратор), в результате чего появится окно, показанное на рис. 2.

****

*Рис. 2. Окно конфигуратора*

Рабочее поле утилиты «Hardware Configuration» разбито на три основные части. В левой верхней части показаны стойки с отдельными слотами. Они расположены на шинах. В левой нижней части находится таблица с адресами входов-выходов, различных блоков и контроллеров. В правой части окна расположена библиотека элементов, из которых можно собирать стойки (рис. 3).

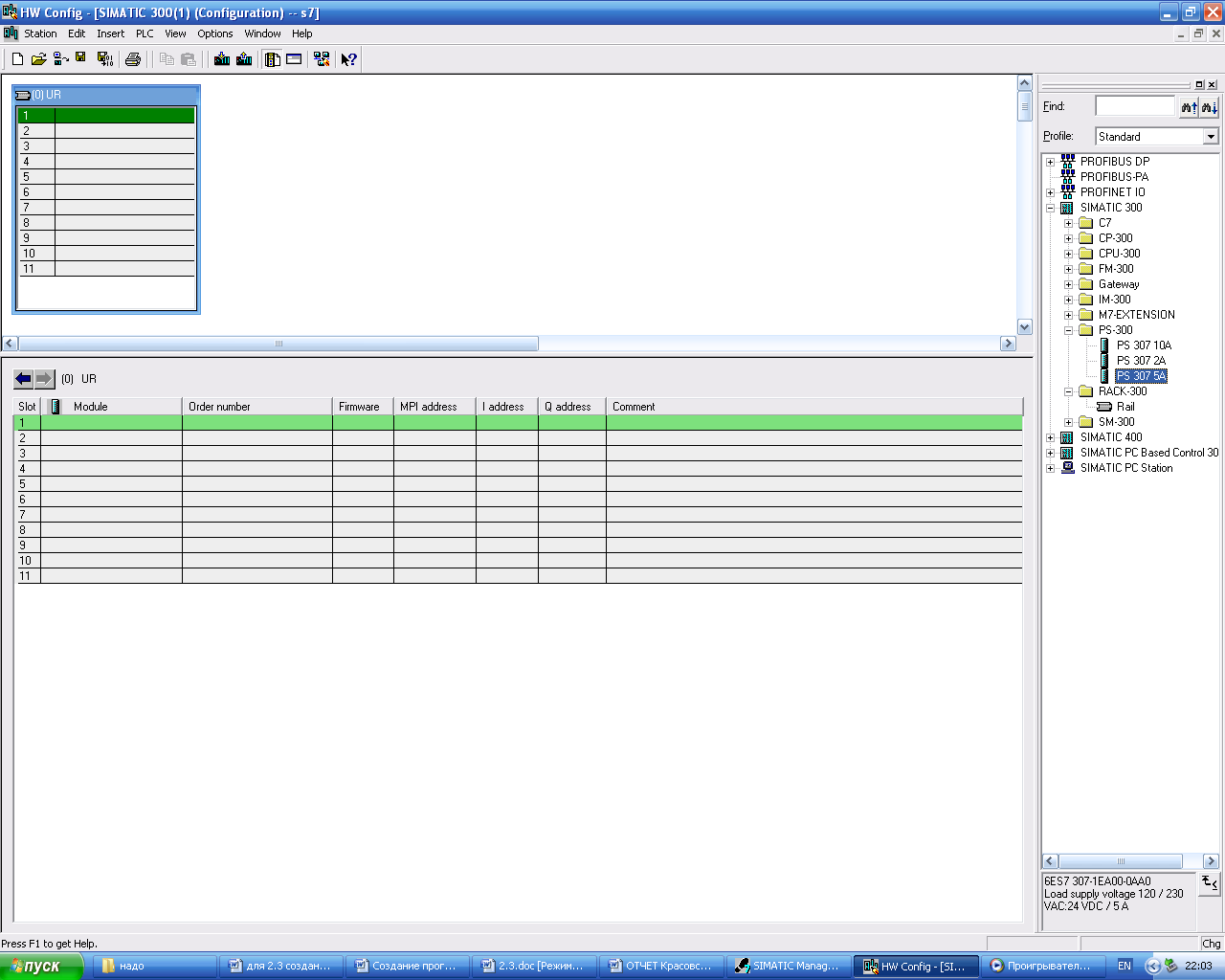


*Рис. 3. Аппаратура, служащая для конфигурирования контроллера*

Создание аппаратной части начинается с добавления стойки (Rack), которая находится в соответствующем каталоге. Например, при создании станции SIMATIC 300 необходимо открыть каталог элементов SIMATIC 300 и из папки Rack-300 добавить элемент Rail (шина). Добавление можно производить либо двойным щелчком, либо перетаскиванием.

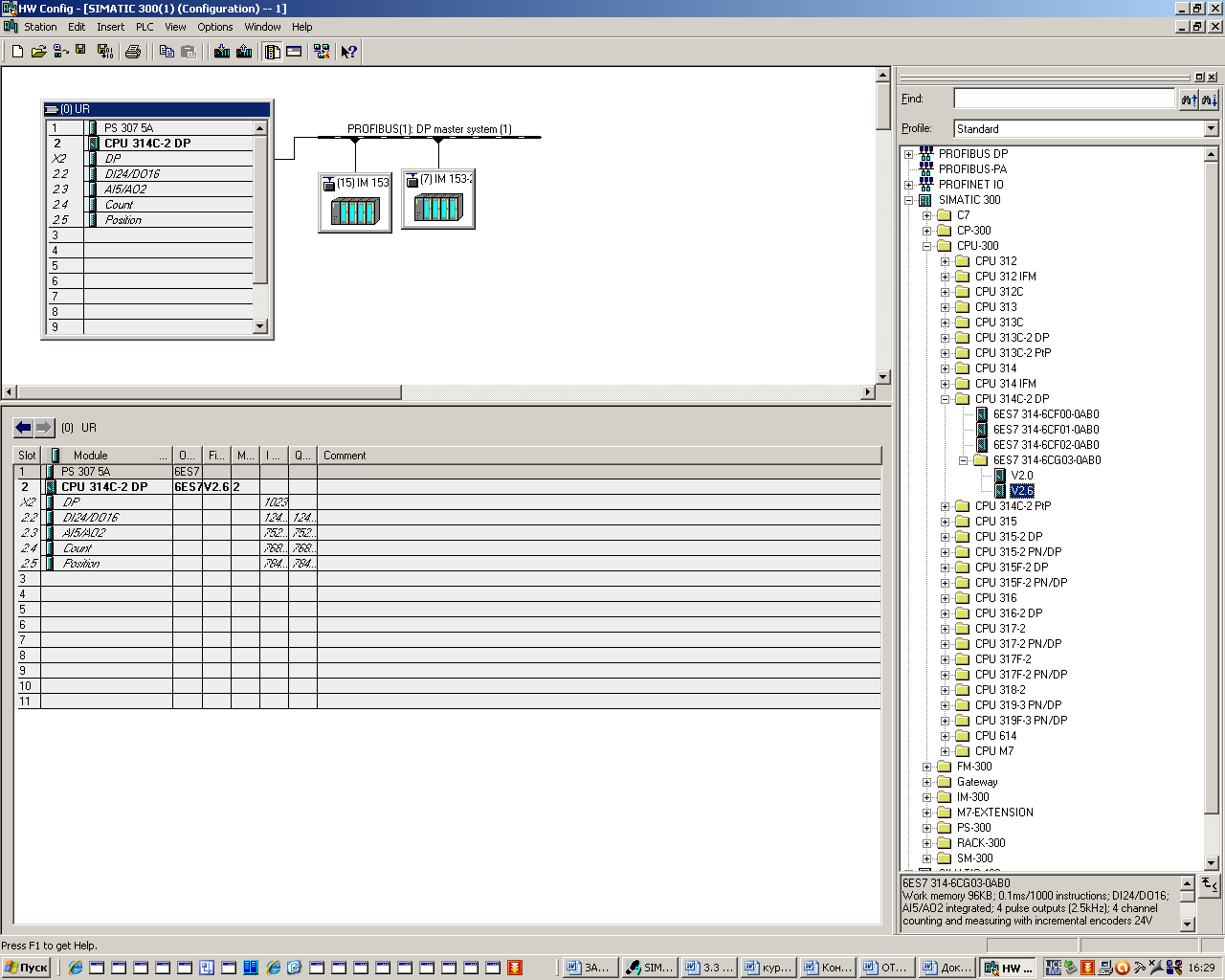
Если требуется установить блок питания, то необходимо вставлять его в слот 1 стойки. Соответствующий модуль станции SIMATIC 300 находится в группе PS-300 (рис. 4).

Если правильно добавлен блок питания, то курсор переходит на вторую строку стойки и выделяется синим цветом



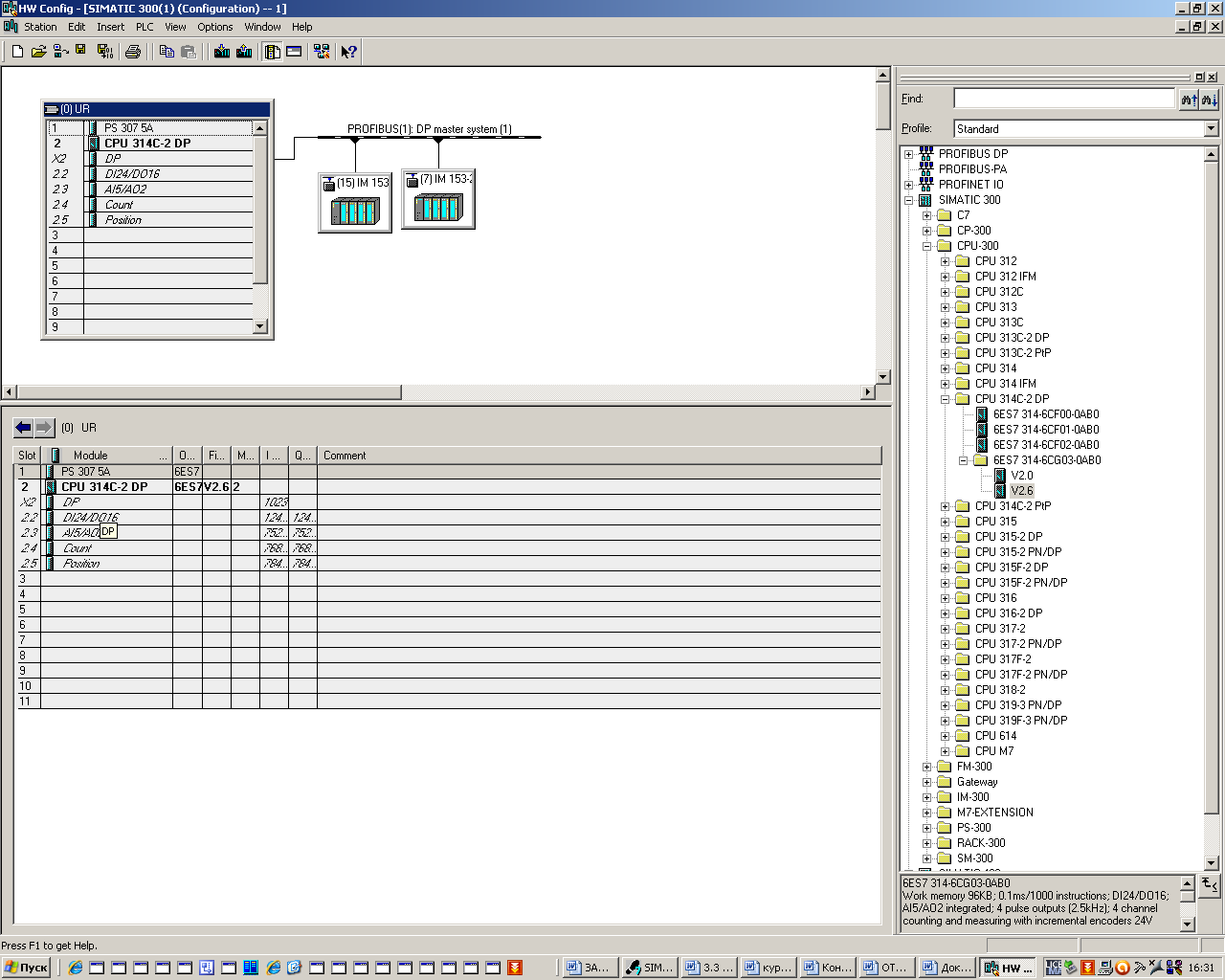
*Рис. 4. Выбор блока питания*

. На второй строке стойки всегда располагается ЦПУ, его можно найти в каталоге CPU-300. Выбираем его также в правой части окна конфигуратора согласно заказному номеру (рис. 5).



*Рис. 5. Выбор ЦПУ*

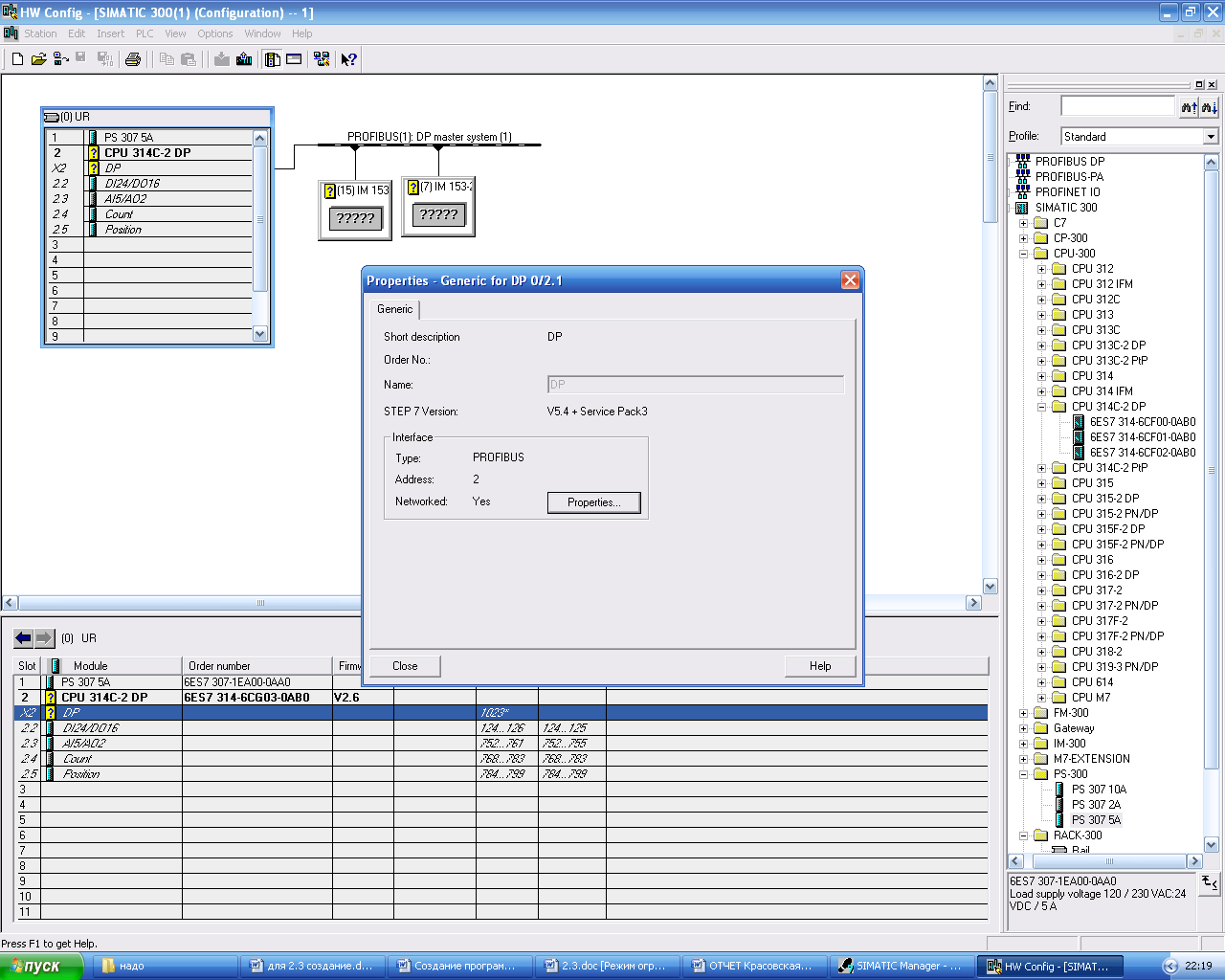
У ЦПУ есть встроенные модули которые при добавлении ЦПУ на стойку (0)UR появляются на строке 2.2, 2.3, Х2, 2.4, 2.5 (рис. 6).



*Рис. 6. Размещение встроенных модулей ЦПУ*

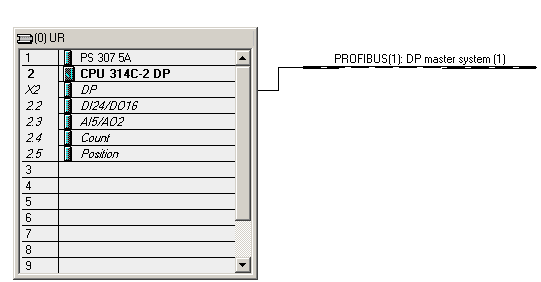
Слот 3 зарезервирован для интерфейсного модуля IM, необходимого для многоуровневых конфигураций.

При добавлении ЦПУ появится окно, показано на рис. 7. В этом окне нужно настроить связь с контроллером посредство PROFIBUS DP. При нажатии кнопки Properties (Свойства) во вкладке параметры нужно выбрать PROFIBUS(1), а также адрес соединения.



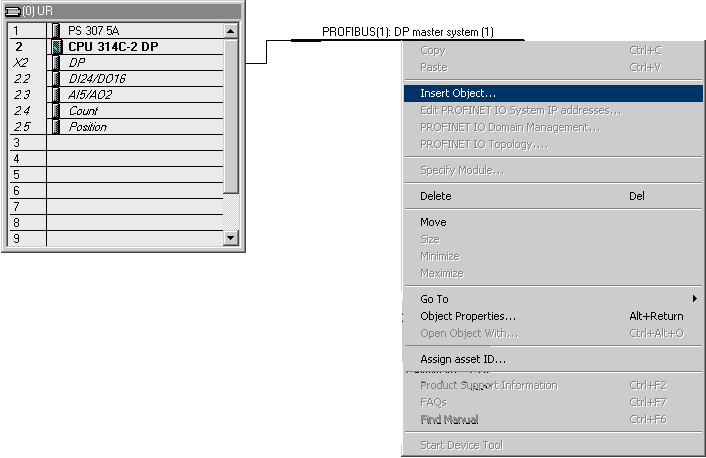
*Рис. 7. Окно свойств DP*

У стойки (0) UR появляется связь с PROFIBUS DP (рис. 8).



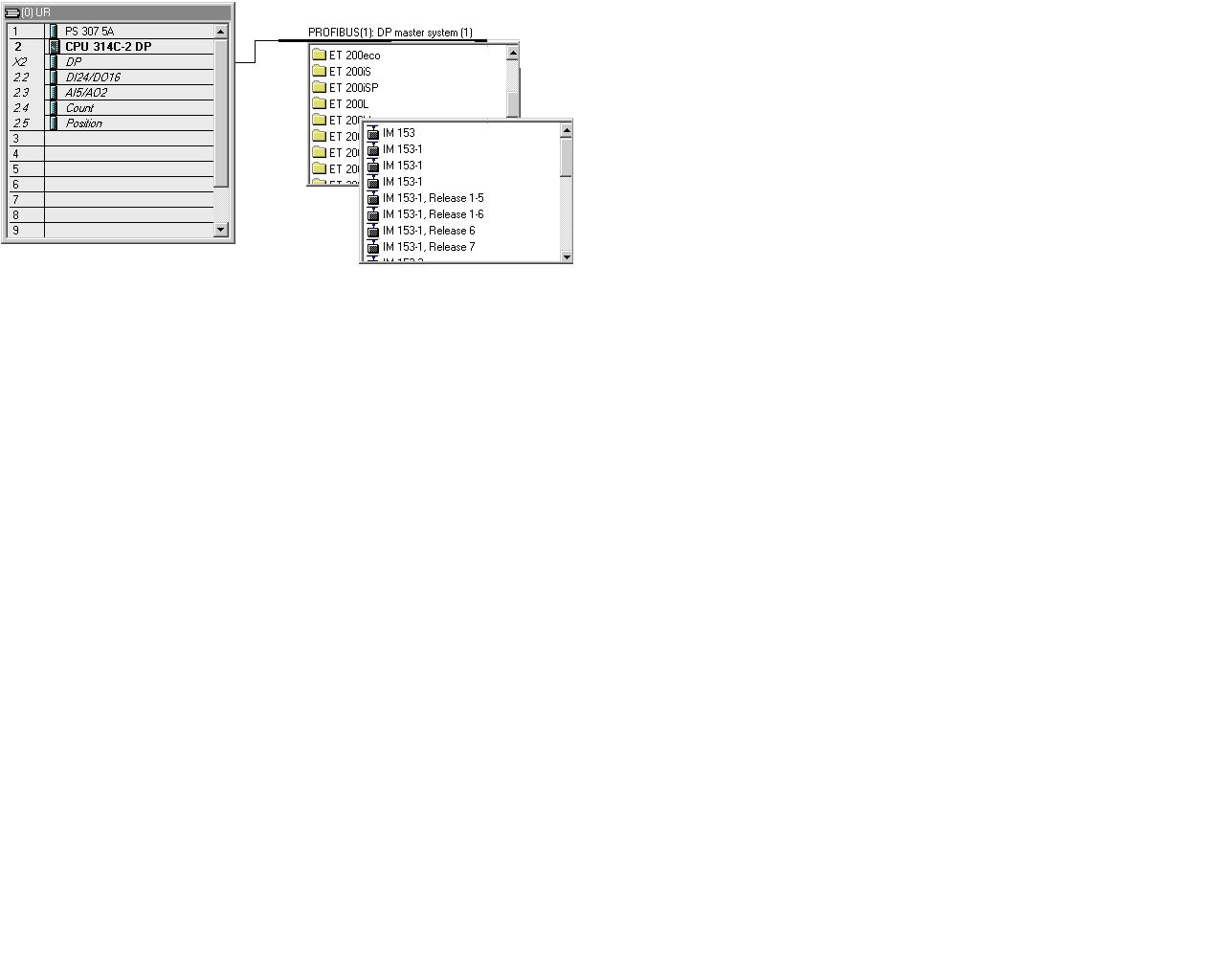
*Рис. 8. Подключение PROFIBUS DP*

Теперь на PROFIBUS(1): DP master system (1) надо "подвесить" распределенные станции ввода-вывода ET 200 M и их интерфейсные модули IM 153. Нажимаем правой кнопкой мыши на PROFIBUS(1): DP master system (1), интерфейс выделится черным цветом. Во всплывающем меню выберем "Insert Object" – "Добавить объект" (рис. 9).



*Рис. 9. Добавление распределенных станций ET 200М*

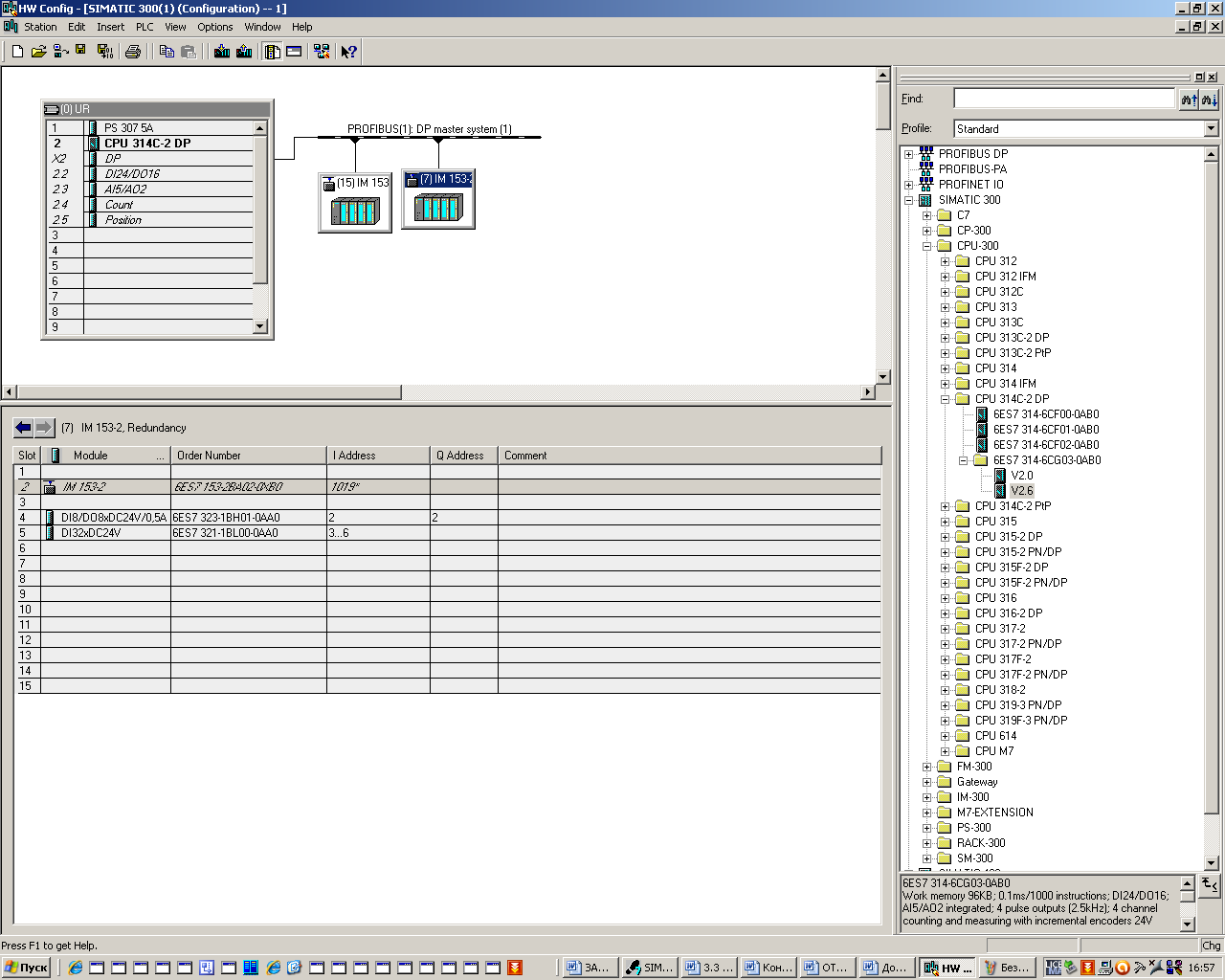
Затем появится окно, где представлены все существующие станции распределенного ввода-вывода (рис. 10).



*Рисунок 10 – Выбор станции ET 200 M*

В этом окне выбираем ET 200М. В новом всплывающем окне выбираем интерфейсные модули IM 153-2, в соответствии с заказным номером указанном на контроллере (рис. 11).

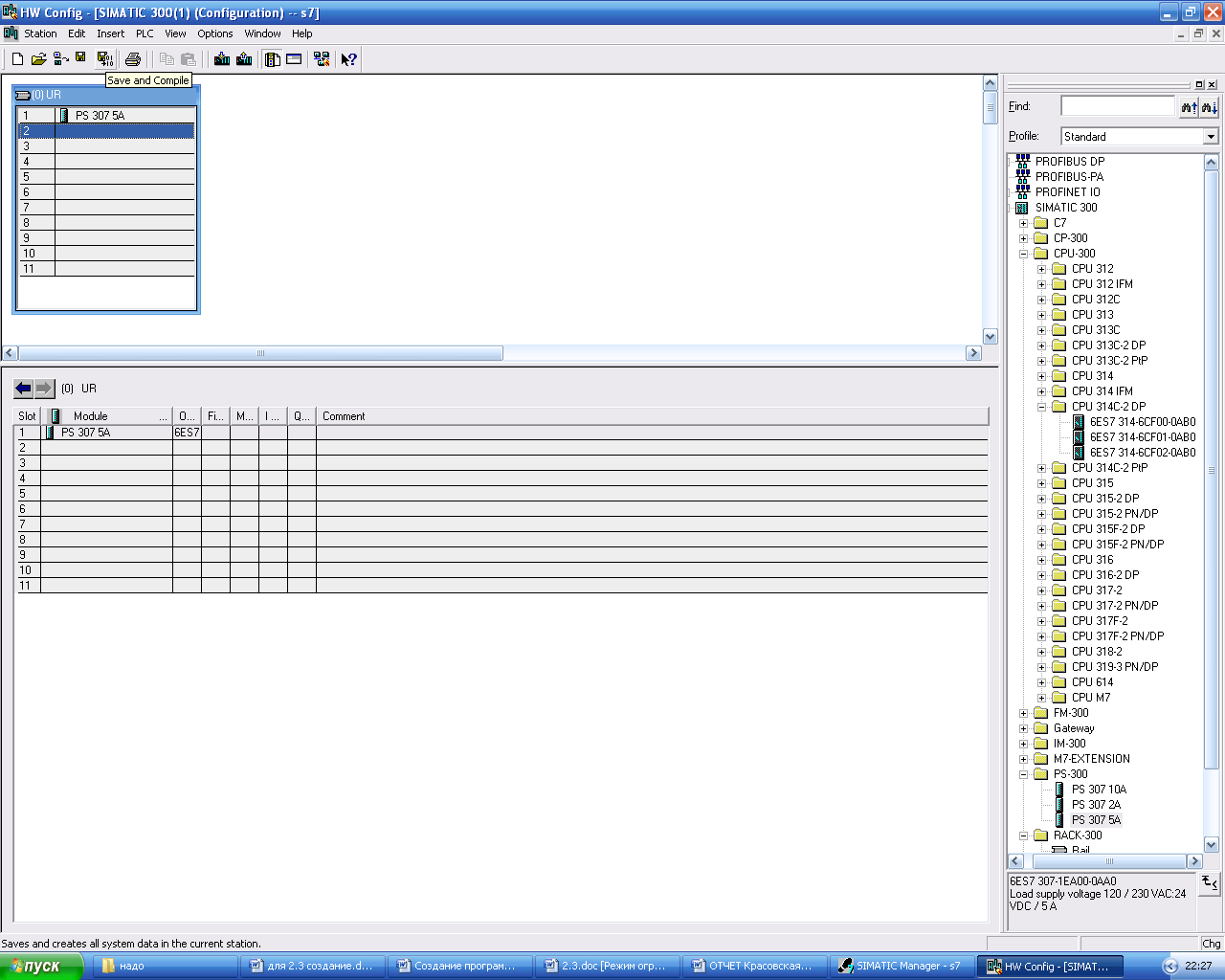
В окне свойств PROFIBUS interface IM 153-2 в первом модуле указываем адрес 7, во втором модуле 15.



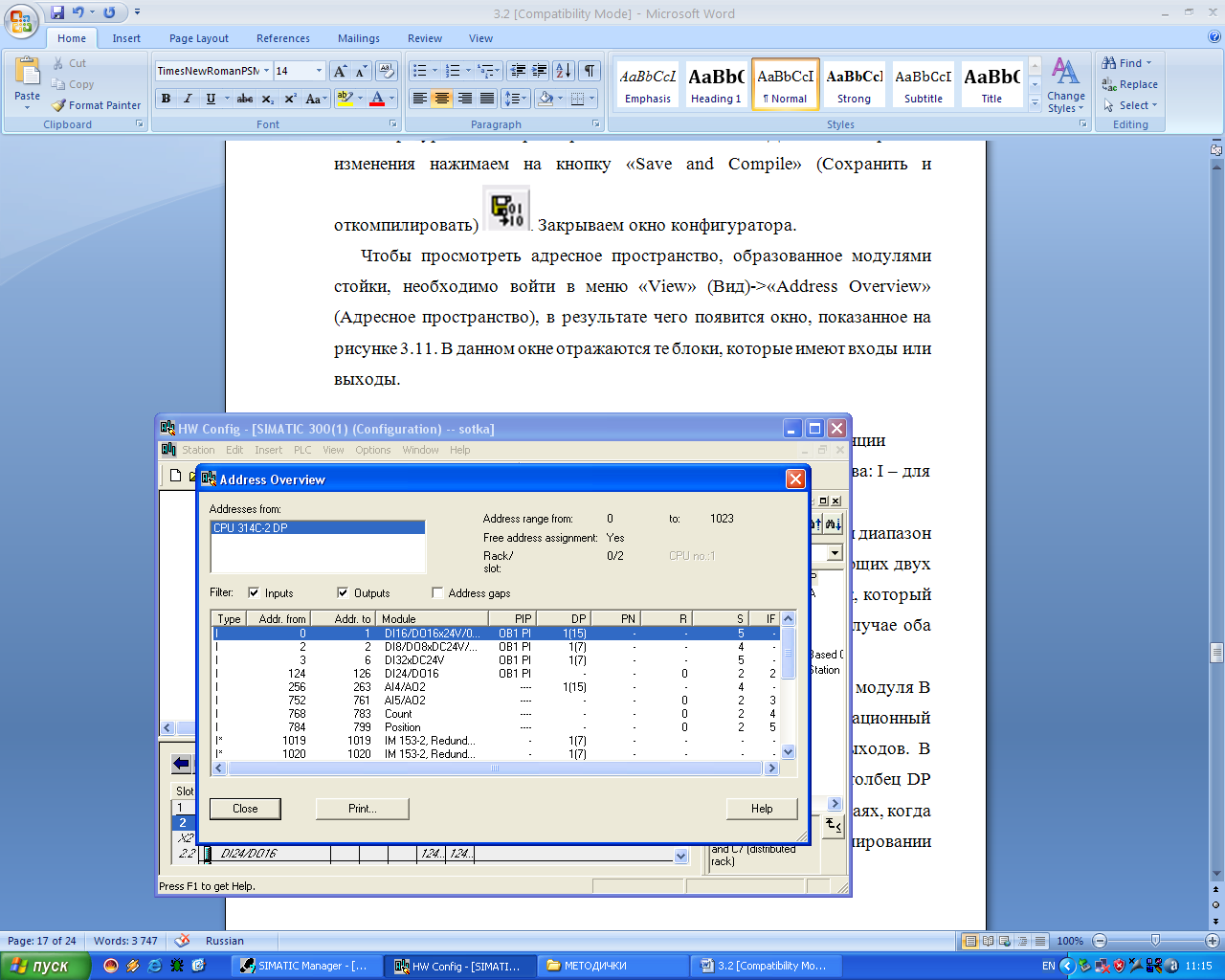
*Рисунок 11 – Добавление модулей*

Начиная с четвертого слота, можно вставлять сигнальные модули. Можно добавить на выбор до 8 сигнальных блоков (SM), коммуникационных процессоров (CP) или функциональных модулей (FM). Необходимо отыскивать нужный модуль в папке и вставлять его, выбирая слот в стойке.

В стандартной конфигурации в стойку может входить процессор, блок питания и модули ввода и вывода, которые бывают аналоговые или дискретные.

Конфигурация контроллера на этом закончина. Для того сохранить изменения нажимаем на кнопку «Save and Compile» (Сохранить и откомпилировать) . Закрываем окно конфигуратора.

Чтобы просмотреть адресное пространство, образованное модулями стойки, необходимо войти в меню «View» (Вид)->«Address Overview» (Адресное пространство), в результате чего появится окно, показанное на рис. 12. В данном окне отражаются те блоки, которые имеют входы или выходы.



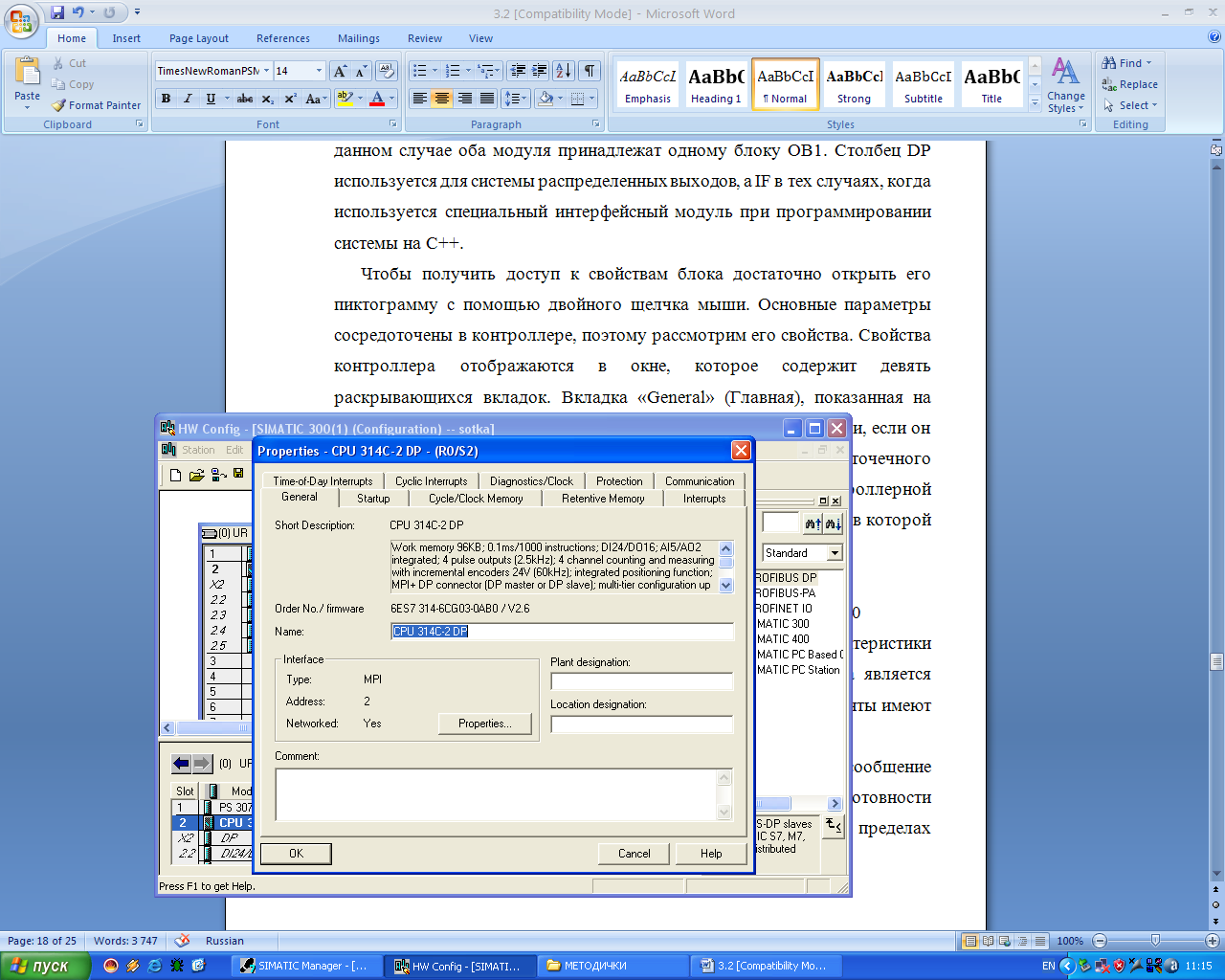
*Рисунок 12 - Адресное пространство аппаратуры станции*

В первом столбце Type указывается тип адресного пространства: I – для входов, Q – для выходов.

Во втором и третьем столбцах Addr. from и Addr. to указывается диапазон адресов в байтах, который занимает данное устройство. В следующих двух столбцах указываются названия блоков и организационный блок, который осуществляет опрос входов и назначение выходов. В данном случае оба модуля принадлежат одному блоку OB1.

Столбец R отображает номер стойки, а столбец S – слота для модуля В следующих двух столбцах указываются названия блоков и организационный блок, который осуществляет опрос входов и назначение выходов. В данном случае оба модуля принадлежат одному блоку OB1. Столбец DP используется для системы распределенных выходов, а IF в тех случаях, когда используется специальный интерфейсный модуль при программировании системы на C++.

Чтобы получить доступ к свойствам блока достаточно открыть его пиктограмму с помощью двойного щелчка мыши. Основные параметры сосредоточены в контроллере, поэтому рассмотрим его свойства. Свойства контроллера отображаются в окне, которое содержит девять раскрывающихся вкладок. Вкладка «General» (Главная), показанная на рис. 13, содержит информацию о типе модуля, его название и, если он программируемый, MPI адрес. Чтобы назначить адрес многоточечного интерфейса, например, в случае создания многоконтроллерной конфигурации, достаточно нажать кнопку «Properties» (Свойства), в которой имеется возможность задать параметр «Adress» (Адрес).



*Рис. 13. Общие параметры контроллера S7-300*

Вкладка «Startup» (Запуск), которая позволяет задавать характеристики запуска. Для S7-300 единственным возможным типом запуска является «Warm restart» («теплый перезапуск»). Только некоторые варианты имеют вариант «Cold restart» («холодный перезапуск»).

Параметр «Finished Message by Modules (ms)» (Законченное сообщение модулей) означает максимальное время на получение сигнала готовности модулей. Если модули не подтверждают приема параметров в пределах установленного времени, то реальная конфигурация не соответствует проектной.

Параметр «Transfer of Parameters to Modules» (Передача Параметров к Модулям) – максимальное время для передачи параметров в настраиваемые модули, после того как получен сигнал готовности.

Параметр «Startup when expected/actual configuration differ» (Запуск, когда ожидаемая/фактическая конфигурация отличаются) позволяет для контроллеров со встроенным интерфейсом распределенных входов-выходов DP и для S7-400 запретить или разрешить запуск, если реальная конфигурация оборудования отличается от проектной. Остальные контроллеры запускаются в любом случае.

Также для контроллеров S7-400 можно указать сбрасывать выходы при горячем перезапуске – «Reset outputs at hot restart» (Продукции сброса в горячем перезапуске) и запретить перезапуск от другой станции или оператора.

Закладка «Retentive Memory» (сохраняемая память) используется для определения областей памяти, которые должны сохраняться после пропадания питания или переходе процессора из режима STOP в RUN. В обоих случаях в S7-300 выполняется полный перезапуск, при котором блоки (OB, FC, FB, DB), хранимые в памяти с батарейной подпиткой, а также меркеры, таймеры и счетчики, определенные как сохраняемые, не изменяются.

Закладка «Cycle/Clock Memory» (Память Цикла/Часов) позволяет с помощью параметра «Scan Cycle Monitoring Time (ms)» (Цикл Просмотра, Контролирующий Время (миллисекунда)) задавать время контроля цикла. Если это время превышено, то контроллер переходит в режим STOP. Возможными причинами превышения времени могут быть коммуникационные процессы, часто от событий прерываний, ошибки в программе.

Параметр «Cycle Load from Communication (%)» (вес цикла от Коммуникации) задает время связи, например, время передачи данных в другой контроллер через многоточечный интерфейс. Это время ограничивается значением, выраженным в процентах от текущего времени цикла.

Например, ограничение связи до 20% приведет к тому, что для времени цикла сканирования 100 мс максимальное время для связи составит 20 мс.

Для синхронизации работы программы используется синхробайт «Clock Memory» (Память Часов), который является байтом из области меркеров. Его биты периодически изменяют свое значение, причем каждый бит в синхробайте связан с конкретной частотой.

Во вкладке «Protection» (Защита) можно изменять параметры защиты. Вкладка «Protection» (Защита) позволяет задать три уровня защиты. На первом уровне «Keyswitch setting» (Установка Keyswitch) можно работать без ограничений. Если назначен пароль, то он определяет следующие ограничения: для уровня 1 – в режиме останова (STOP) возможен полный доступ, а в режиме работы (RUN) только чтение; для уровня защиты 2 – существует доступ только для чтения, а для уровня 3 – невозможно ни чтение, ни запись или 3, либо выбрать режим «Removable with password» (Сменный с паролем) на первом уровне.

Вкладка «Diagnostic/Clock» (Диагностика/ Часы) позволяет с помощью флага «Report cause of stop» (Причина сообщения остановки) обнаруживать причину останова, а также синхронизировать часы нескольких контроллеров и вводить коррекцию для часов.

Кроме перечисленных, также имеются вкладки, определяющие параметры прерываний «Interrupts», «Cyclic Interrupts», «Time-of-Day Interrupts».

Вкладка «Interrupts» (перерывы): приоритеты программ задаются по возрастанию, т.е. чем выше номер, тем более высокий приоритет. Для циклических прерываний имеется возможность указать интервал выполнения через параметр «Execution» (Выполнение), а для прерываний, вызываемых по времени суток, указываются параметры «Start Date» (Дата начала) и «Time of Day» (Дата/время).

Чтобы сохранить конфигурацию, нужно войти в меню «Station» (Станция) и выбрать вкладку «Save» (Сохранить). При выборе вкладки «Save and Compile» (Сохранить и откомпилировать) конфигурация загружается в блоки данных DB проекта. Чтобы проверить правильность конфигурации, можно воспользоваться меню «Station» (Станция) -> «Consistency Check» (Проверка на непротиворечивость). Загрузка конфигурации в контроллер или его эмулятор возможна через меню, «PLC» (ПЛК) -> «Download» (Загрузить), при этом контроллер должен находиться в режиме STOP.

Для входных и выходных модулей можно задавать их адреса, однако необходимо помнить, что после перезапуска контроллера снова применяется адресация по умолчанию.

**8. Настройка интерфейсов**

**8.1. Основные понятия об интерфейсах**

Различают следующие соединения с помощью MPI, PROFIBUS, ASI и т.п.

**Message Passing Interface** (MPI, интерфейс передачи сообщений)  -программный интерфейс ([API](http://ru.wikipedia.org/wiki/API)) для передачи [информации](http://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F), который позволяет обмениваться сообщениями между процессами, выполняющими одну задачу. Разработан [Уильямом Гроуппом](http://en.wikipedia.org/wiki/William_Gropp), [Эвином Ласком](http://en.wikipedia.org/wiki/Ewing_Lusk) и другими.

MPI является наиболее распространённым стандартом интерфейса обмена данными в [параллельном программировании](http://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D0%BF%D1%80%D0%B5%D0%B4%D0%B5%D0%BB%D1%91%D0%BD%D0%BD%D1%8B%D0%B5_%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F), существуют его реализации для большого числа компьютерных платформ. Используется при разработке программ для [кластеров](http://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%82%D0%B5%D1%80_(%D0%B3%D1%80%D1%83%D0%BF%D0%BF%D0%B0_%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BE%D0%B2)) и [суперкомпьютеров](http://ru.wikipedia.org/wiki/%D0%A1%D1%83%D0%BF%D0%B5%D1%80%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80). Основным средством коммуникации между [процессами](http://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) в MPI является передача сообщений друг другу. Стандартизацией MPI занимается MPI Forum. В стандарте MPI описан интерфейс передачи сообщений, который должен поддерживаться как на платформе, так и в [приложениях](http://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) пользователя.

**AS-Interface** – [промышленная сеть](http://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%BC%D1%8B%D1%88%D0%BB%D0%B5%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C), предназначенная для передачи преимущественно дискретных сигналов. Используется обычно в [машиностроении](http://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BE%D1%81%D1%82%D1%80%D0%BE%D0%B5%D0%BD%D0%B8%D0%B5). Является «открытой» технологией. Спецификация разработана и поддерживается ведущими производителями систем автоматизации (в настоящий момент свыше 100 фирм-участниц). Топология сети — любая. Для подключения датчиков разработан специальный плоский кабель с подключением под прокол изоляции (ножевые клеммы предусмотрены в конструкции модулей ввода-вывода). Версия AS-i 2.0 позволяет передавать аналоговые сигналы. Новейшей версией спецификации является AS-i 3.0.

Существует профиль протокола для систем повышенной безопасности ASi-Safe. Устройства повышенной безопасности подключаются по тому же кабелю и поддерживают уровень безопасности вплоть до SIL ([Safety Integrity Level](http://ru.wikipedia.org/w/index.php?title=Safety_Integrity_Level&action=edit&redlink=1)) 3 согласно IEC 61508 и вплоть до Safety Category 4 согласно EN 954-1.

**PROFIBUS** (Process Field Bus) - открытая [промышленная сеть](http://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%BC%D1%8B%D1%88%D0%BB%D0%B5%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C), прототип которой был разработан компанией [Siemens AG](http://ru.wikipedia.org/wiki/Siemens_AG) для своих промышленных контроллеров [SIMATIC](http://ru.wikipedia.org/wiki/SIMATIC), на основе этого прототипа Организация пользователей PROFIBUS разработала международные стандарты, принятые затем некоторыми национальными комитетами по стандартизации. Очень широко распространена в [Европе](http://ru.wikipedia.org/wiki/%D0%95%D0%B2%D1%80%D0%BE%D0%BF%D0%B0), особенно в машиностроении и управлении промышленным оборудованием. Сеть PROFIBUS — это комплексное понятие, она основывается на нескольких стандартах и протоколах. Сеть отвечает требованиям международных стандартов [IEC 61158](http://ru.wikipedia.org/w/index.php?title=IEC_61158&action=edit&redlink=1) и [EN 50170](http://ru.wikipedia.org/w/index.php?title=EN_50170&action=edit&redlink=1). Поддержкой, стандартизацией и развитием сетей стандарта PROFIBUS занимается PROFIBUS NETWORK ORGANISATION (PNO).

PROFIBUS объединяет технологические и функциональные особенности последовательной связи полевого уровня. Она позволяет объединять разрозненные устройства автоматизации в единую систему на уровне датчиков и приводов.

PROFIBUS использует обмен данными между [ведущим](http://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B4%D1%83%D1%89%D0%B5%D0%B5_%D1%83%D1%81%D1%82%D1%80%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%BE) и [ведомыми устройствами](http://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B4%D0%BE%D0%BC%D0%BE%D0%B5_%D1%83%D1%81%D1%82%D1%80%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%BE) (протоколы [DP](http://ru.wikipedia.org/wiki/PROFIBUS_DP) и [PA](http://ru.wikipedia.org/wiki/PROFIBUS_PA)) или между несколькими ведущими устройствами (протоколы [FDL](http://ru.wikipedia.org/w/index.php?title=PROFIBUS_FDL&action=edit&redlink=1) и [FMS](http://ru.wikipedia.org/wiki/PROFIBUS_FMS)). Требования пользователей к получению открытой, независимой от производителя системе связи, базируется на использовании стандартных протоколов PROFIBUS.

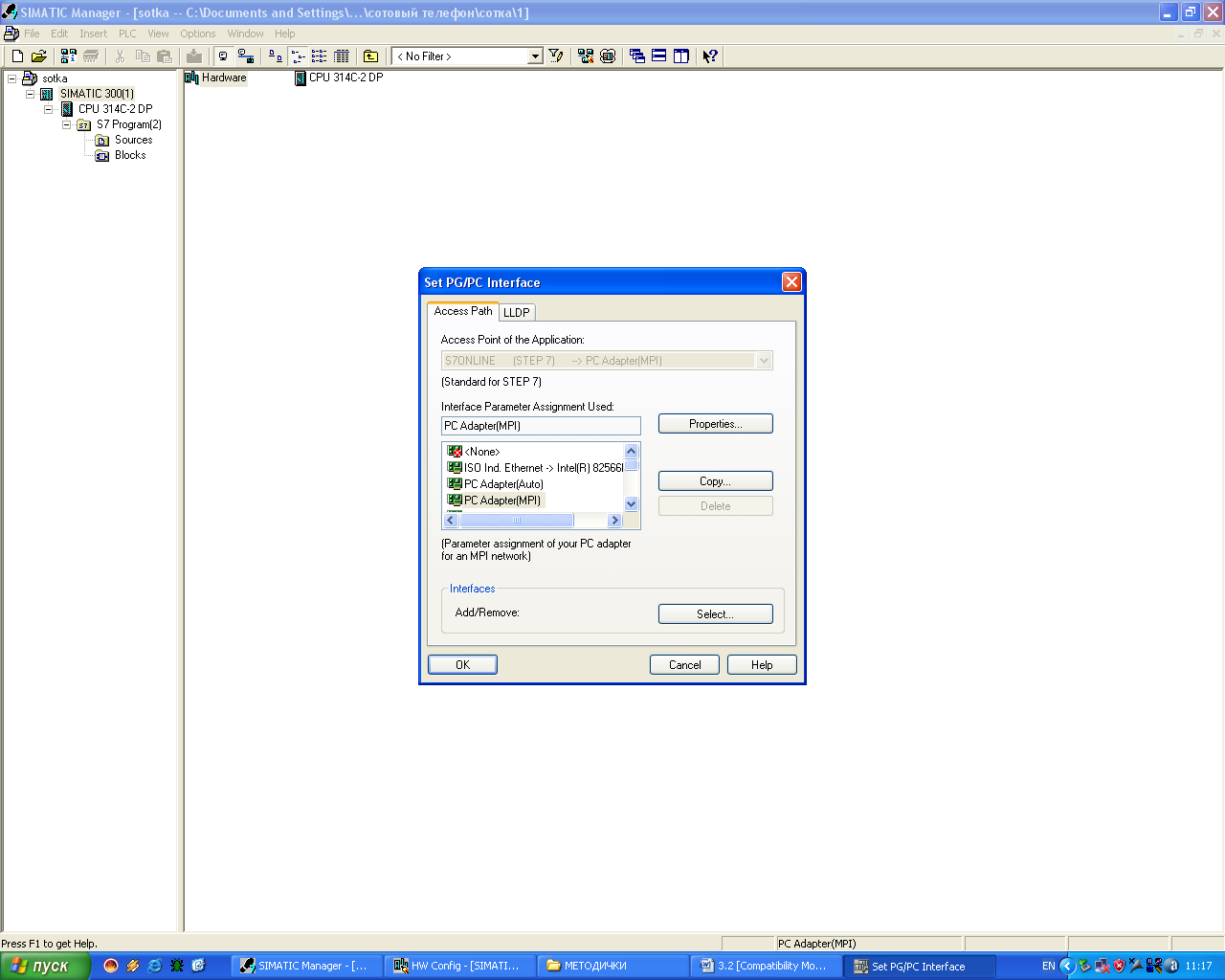
Сеть PROFIBUS построена в соответствии с [многоуровневой сетевой моделью ISO 7498](http://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_OSI)

.

**8.2. Настройка MPI и Ethernet**

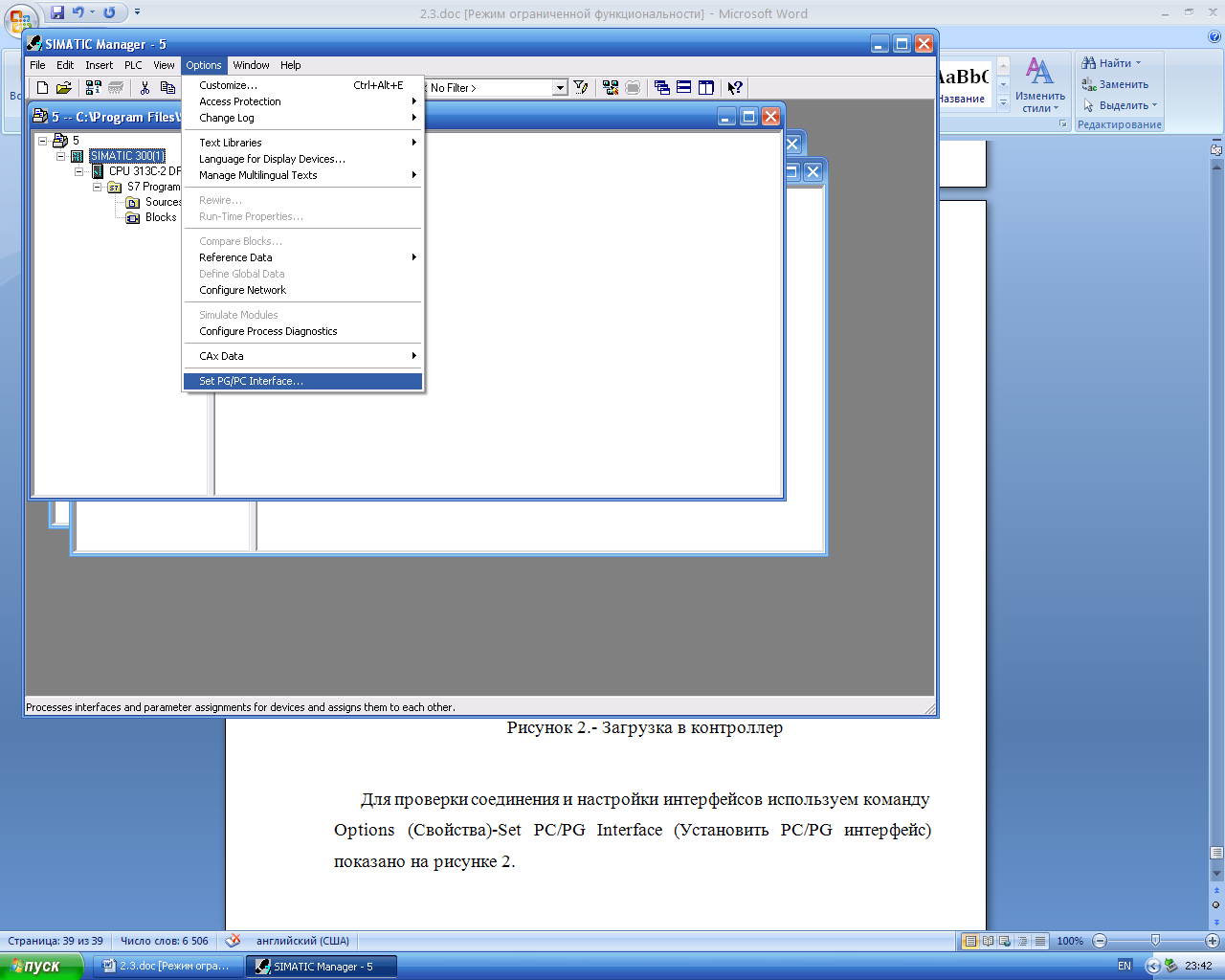
В базовой комплектации контроллера Simatic S7-300 имеется несколько интерфейсов - MPI, Ethernet. Адрес MPI интерфейса - 2. Предустановленная скорость передачи у всех CPU составляет 187,5 Кбит/с. Максимальная скорость передачи составляет 12 Мбит/с.

Для работы интерфейса необходимо настроить РС адаптер. Различают два способа настройки РС адаптера: при помощи меню Siemens Automation - Simatic – Step 7 – Set PG/PC (рис. 14).



*Рисунок 14 – Настройка PC адаптера для интерфейса MPI*

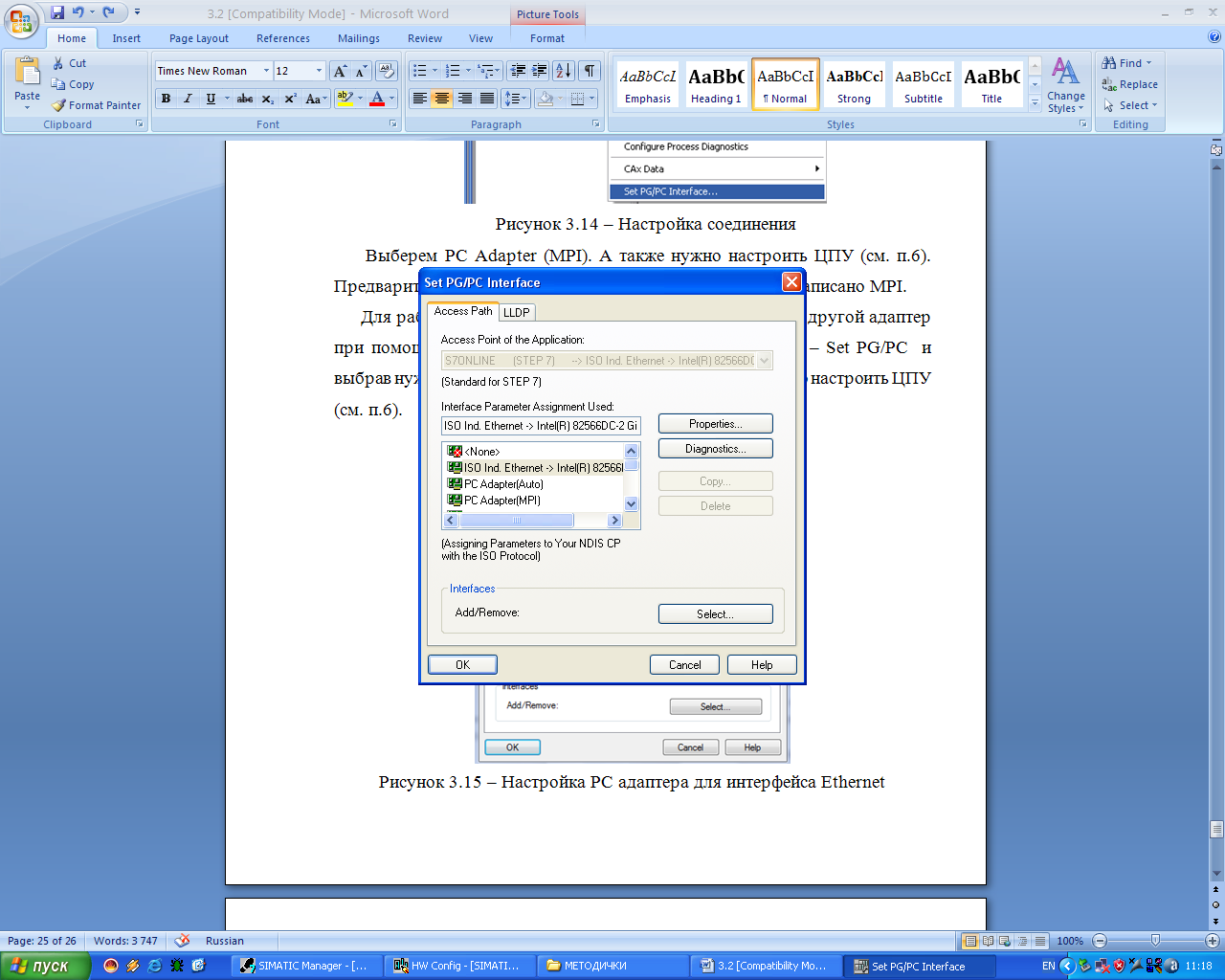
Второй способ настройки связи: «Options» (Опции) – «Set PG/PC» (Установить PG/PC интерфейс), показано на рис. 15.



*Рис. 15. Настройка соединения*

Выберем PC Adapter (MPI). А также нужно настроить ЦПУ (см. п.6). Предварительно нажав двойным щелчком на строке, где написано MPI.

Для работы интерфейса Ethernet необходимо настроить другой адаптер при помощи меню Siemens Automation - SIMATIC – Step 7 – Set PG/PC и выбрав нужный интерфейс (рис.16). Затем необходимо настроить ЦПУ (см. п.7).



*Рис. 16. Настройка PC адаптера для интерфейса Ethernet*

**Задание на лабораторную работу**

1. Сконфигурировать аппаратные средства; 2. Настроить интерфейсы.

*Методические указания по выполнению работы*

В соответствии с перечисленным выше пунктом 7 сконфигурировать аппаратные средства контроллера, а также настроить интерфейсы.

**Содержание отчета**

- титульный лист;

- цель работы;

- структурную схему лабораторного стенда;

- структуру центральной стойки после конфигурирования аппаратуры;

- ответы на контрольные вопросы;

- выводы по работе.

**Контрольные вопросы:**

1. Дайте определение STEP 7?

2. Назовите основные элементы STEP 7?

3. Из чего состоит SIMATIC Manager?

4. Перечислите основные языки программирования STEP 7? В чем различие между ними?

5. Перечислите основные этапы конфигурирования?

Учебное издание

СКОРОСПЕШКИН Владимир Николаевич

**КОНФИГУРИРОВАНИЕ АППАРАТНЫХ СРЕДСТВ КОНТРОЛЛЕРА SIMATIC S7-300**

Методические указания к выполнению лабораторной работы по курсу «Автоматизированные информационно-управляющие системы» для студентов IV курса, обучающихся по направлению 15.03.04 «Автоматизация технологических процессов и производств»

**Зарегистрировано в Издательстве ТПУ**

**Размещено на корпоративном портале ТПУ**

**В полном соответствии с качеством представленного оригинал-макета**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

