

УТВЕРЖДАЮ

Начальник службы автоматизации

ООО «НПП ОЗНА-Инжиниринг»

\_\_\_\_\_\_\_\_\_\_В.О. Литвинов

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

**Система измерений количества и**

**показателей качества нефти**

**№102 ПСП «Тайшет-2»**

**ООО «Транснефть - Восток»**

ОИ 410-01.00.00.00.000 И33.5

**Руководство программиста среднего уровня.**

**Руководство программиста. Часть 5**

РАЗРАБОТАЛ

Главный специалист по РПО МСО

ООО «НПП ОЗНА-Инжиниринг»

\_\_\_\_\_\_\_\_\_\_\_А.С. Русских

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

Уфа 2020

# СОДЕРЖАНИЕ

|  |
| --- |
| Подпись и дата |

|  |
| --- |
| Инв. № дубл. |

|  |
| --- |
| Взам. инв. № |

|  |
| --- |
| Подпись и дата |

|  |
| --- |
| Инв. № подл. |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

Изм.

Лист

№ докум.

Подпись

Дата

Лист

2

**ОИ 410-01.00.00.00.000 И33.5**

Разраб.

А.С. Русских

Провер.

А.А. Решетников

Н. Контр.

*М.В. Самкова*

Утверд.

В.О. Литвинов

Руководство программиста среднего уровня

Лит.

Листов

30

**ООО «НПП ОЗНА-Инжиниринг»**

[СОДЕРЖАНИЕ 2](#_Toc14182982)

[1 ВВЕДЕНИЕ 3](#_Toc14182983)

[2 ОБЩИЕ СВЕДЕНИЯ 4](#_Toc14182984)

[3 СРЕДА РАЗРАБОТКИ STEP 7 5](#_Toc14182985)

[3.1 Общие сведения 5](#_Toc14182986)

[3.2 Операционная система реального времени SIMATIC Manager 6](#_Toc14182987)

[3.3 Начало работы 7](#_Toc14182988)

[3.4 Конфигурирование программного обеспечения 7](#_Toc14182989)

[3.5 Конфигурирование оборудования 8](#_Toc14182990)

[3.6 Выполнение сборки (Build) конфигурации 9](#_Toc14182991)

[3.7 Таблица символов 10](#_Toc14182992)

[3.8 Сохранение проекта 10](#_Toc14182993)

[3.9 Соединение с ПЛК 10](#_Toc14182994)

[3.9.1 Подключение программатора к CPU 10](#_Toc14182995)

[3.9.2 Проверка актуальности проекта в CPU 11](#_Toc14182996)

[3.9.3 Тестирование сигналов ввода вывода из утилиты HW Config 11](#_Toc14182997)

[3.10 Резервное копирование 12](#_Toc14182998)

[3.11 Восстановление из резервной копии 16](#_Toc14182999)

[4 Введение в программирование в среду разработки SIMATIC Manager 20](#_Toc14183000)

[4.1 Общие сведения 20](#_Toc14183001)

[4.2 Инициализация и циклические части программы 20](#_Toc14183002)

[4.3 Переменные и типы данных 21](#_Toc14183003)

[4.4 Символьная адресация переменных 23](#_Toc14183004)

[ПРИЛОЖЕНИЕ А 26](#_Toc14183005)

[Лист регистрации изменений 30](#_Toc14183006)

Н.Л. Елизарьева

*М.В. Самкова*

# ВВЕДЕНИЕ

Настоящее руководство содержит описание и сведения, необходимые для понимания принципов построения и корректировки программного обеспечения программируемого логического контроллера (ПЛК), входящего в состав СОИ СИКН.

Пользователь, осуществляющий процесс корректировки ПО должен обладать необходимым уровнем знаний и навыков для работы с прикладным программным обеспечением, иметь общее представление о системе в целом и о работе на персональной электронно-вычислительной машине (ПЭВМ), уметь пользоваться настоящим руководством и инструкциями служб предприятия.

# ОБЩИЕ СВЕДЕНИЯ

СОИ СИКН предназначена для автоматизированных измерений, сбора, обработки, отображения и регистрации показателей качества нефти.

Система обеспечивает централизованный контроль состояния объекта, сигнализацию отклонения параметров от нормы, дистанционное управление работой объекта, формирование журнала аварийных и технологических сообщений, формирование и печать отчетных документов, ведение базы данных.

ПЛК является неотъемлемой частью системы и выполняет следующие функции:

а) сбор и обработка информации с датчиков, агрегатов и других узлов системы;

б) выполнение команд управления, полученных от АРМ оператора;

в) выполнение алгоритмов защиты и управления исполнительными механизмами системы;

г) передача информации по каналам связи в АРМ оператора.

ПЛК СОИ СИКН выполнен на базе контроллера S7-400H производства Siemens. Для разработки ПО контроллеров данной серии используется программный пакет «STEP 7».

# СРЕДА РАЗРАБОТКИ STEP 7

## Общие сведения

STEP 7 представляет собой пакет стандартного программного обеспечения, используемый для конфигурирования и программирования программируемых логических контроллеров SIMATIC. Он является частью промышленного программного обеспечения SIMATIC.

STEP 7 позволяет решать задачи по созданию и управлению проектами. Среда программирования поддерживает несколько языков программирования, а также ряд диагностических средств и редакторов. Благодаря наличию стандартных библиотек, разработанных компанией Siemens, и интегрированной в систему поддержке языков программирования МЭК, обеспечивается высокая эффективность рабочего процесса.

В программное обеспечение STEP 7 встроен широкий инструментарий, позволяющий решить практически любую прикладную задачу. Полно и детально описанные ответы на многие вопросы по программному и аппаратному обеспечению можно найти в справочной системе STEP 7.

При решении задачи автоматизации STEP 7 обеспечивает решение таких задач, как:

* Создание и управление проектами;
* Конфигурирование и назначение параметров аппаратуре и связям;
* Управление символами;
* Создание программ, например, для программируемых контроллеров S7;
* Загрузка программ в программируемые контроллеры;
* Тестирование системы автоматизации;
* Диагностика неисправностей установки.

Ядром программного обеспечения котроллера является операционная система реального времени SIMATIC Manager. Он является неотъемлемым компонентом пакета STEP 7.

## Операционная система реального времени SIMATIC Manager

SIMATIC Manager организован в виде модульной структуры и обеспечивает возможность быстрого повторного выполнения приложения в пределах выделенного интервала времени. В итоге это позволяет достичь оптимальных количественных и качественных характеристик, а также заданной точности во время выполнения программы.

SIMATIC Manager предоставляет пользователю детерминированный, аппаратно-независимый, многозадачный инструмент для создания приложений.

Программе (также называемой задачей) может быть назначено определенное время выполнения или класс задач, в течение которого она выполняется циклически.

Использование функций МЭК-библиотек ускоряет и упрощает разработку, а также помогает избежать ошибок.

SIMATIC Manager полностью интегрирован в соответствующую целевую систему, что позволяет прикладным программам получать доступ к модулям ввода/вывода системы, интерфейсам, полевым шинам, сетям и устройствам хранения данных.

SIMATIC Manager - графический интерфейс для редактирования online и offline объектов S7 (проектов, файлов пользовательских программ, блоков, аппаратной части станций). В SIMATIC Manager Вы можете:

• управлять проектами и библиотеками

• запускать утилиты STEP 7

• подключаться к PLC

• редактировать содержимое модуля памяти.

SIMATIC Manager выполняет ряд важных функций:

а) обеспечивает независимость приложения от аппаратного обеспечения;

б) обеспечивает детерминированное поведение благодаря циклическому алгоритму системы исполнения;

в) позволяет настроить различное время цикла;

г) поддерживает 8 различных классов задач;

д) гарантирует реакцию на превышение времени цикла;

е) обеспечивает настройку предельно допустимых отклонений для всех классов задач;

ж) поддерживает библиотеки функций в соответствии с МЭК 61131-3;

з) обеспечивает доступ к любым сетям и шинам.

Операционная система SIMATIC Manager и приложения, разработанные в SIMATIC Manager, запускаются на целевой системе с карты CompactFlash.

## Начало работы

При установке STEP 7 в ОС Windows в меню «Пуск» и на рабочий стол добавляются новые элементы. SIMATIC Manager может быть запущена из меню «Пуск» или с помощью ярлыка на рабочем столе Windows.

При первом запуске SIMATIC Manager открывается начальное окно, показанное в Приложении А. Данное окно SIMATIC Manager позволяет создать новый проект или открыть существующий.

Для открытия существующего проекта можно воспользоваться меню <File>/<Open…>, либо сочетанием клавиш «Ctrl + O». В появившемся окне нажать «Browse…» поиска проекта необходимо перейти в каталог с проектом СОИ СИКН (изначально находится на диске в комплекте поставки системы) и выбрать файл с проектом.

SIMATIC Manager управляет объектами S7, в частности, проектами и пользовательскими программами. При открытии объекта запускается ассоциированный с ним редактор. Например, двойной щелчок на программном блоке запускает редактор программ, после чего блок может быть отредактирован (объектно-ориентированный запуск).

Вы можете всегда получить контекстную помощь для текущего окна, нажав функциональную клавишу F1.

## Конфигурирование программного обеспечения

STEP 7 включает в себя несколько языков программирования, которые могут использоваться в зависимости от индивидуальных предпочтений и знаний. Программу, созданную на языке списка команд, с соблюдением специфических правил, можно затем преобразовывать в программу на другом языке программирования. В свою очередь, программу, созданную на любом другом языке программирования можно всегда преобразовать в список команд.

LAD – Это контактный план подобен схеме электрической цепи. Используются, например, такие символы, как контакты и катушки. Этот язык программирования рассчитан на тех, кто привык работать с электрическими схемами.

STL – это список инструкций (Statement list) содержит команды языка STEP 7. На STL Вы можете довольно свободно программировать (в частности, можно написать программы, не доступные на других языках STEP 7). Этот язык программирования предпочитают программисты, которые уже знакомы с другими языками программирования.

FBD – это язык функциональных блоков для обозначения конкретных функций использует прямоугольники. Символ в прямоугольнике указывает на функцию (например, & - логическая функция И). Этот язык программирования имеет то преимущество, что с ним может работать даже не программист, например, инженер-технолог. Язык функциональных блоков введен в STEP7 , начиная с версии 3.0.

## Конфигурирование оборудования

HW Config помогает Вам при конфигурировании, назначении параметров и диагностике аппаратуры.

Для запуска утилиты HW Config: в SIMATIC Manager выделите объект-станцию и выберите меню опций <Edit>/<Open Object>.

«Hardware Configuration» окно утилиты «HW Conlig» Вы используете для установки компонентов из окна «Hardware Catalog».

Строка заголовка этого окна содержит имена проекта и станции.

Для открытия каталога: Выберите команду меню <View>/<Catalog>.

Если в качестве профиля каталога (набор компонентов) выбран «Standard», то для выбора предлагаются все стойки, модули и интерфейсные модули. Пользователи могут создавать свои собственные профили каталога, содержащие часто используемые ими элементы, выбирая меню <Options>/<Edit Catalog Profiles>.

Конфигурирование сетевых интерфейсов. Для обмена данными с контроллером среда SIMATIC Manager требует подключения к сети.

Чтобы обеспечить эту возможность, должны быть выполнены следующие условия:

а) ПК и контроллер находятся в одной и той же сети;

б) известны разрешенные и назначенные IP адреса.

Если компьютер уже работает в сети, необходимо перед изменением параметров записать их текущие значения, чтобы позже можно было их восстановить. Для получения более подробной информации о настройке сетевых параметров обратитесь к вашему системному администратору.

Свойства сети, относящиеся к контроллеру, могут быть открыты из контекстного меню соответствующего Ethernet-интерфейса контроллера (см. приложение А).

## Выполнение сборки (Build) конфигурации

Перед тестированием или работой Ваша программа должна быть откомпилирована. Чтобы быть уверенным, что Вы компилируете последнюю сохраненную версию исходного файла, выберите команду меню <Options>/<Customize> (<Параметры>/<Настройка>) и установите опцию «Save before compiling (Сохранять перед компиляцией)» на закладке «Editor (Редактор)». Тогда команда меню <File>/<Compile> (<Файл>/<Компилировать>) сохранит исходный файл SCL.

Проделайте следующие шаги:

1. Сохраните исходный файл SCL перед компиляцией.

2. Для создания S7 программ, Вы должны выбрать опцию «Create Object code (Создание объектной программы)» на закладке «Compiler (Компилятор)» в окне «Customize (Настройка)».

3. Если требуется, установите иные модификации компилятора.

4. Проверьте таблицу символики в соответствующей программной папке.

5. Вы можете начать компиляцию так:

Команда меню <File>/<Compile> (<Файл>/<Компилировать>) компилирует исходный файл целиком.

Команда меню <File>/<Compile Selected Blocks> (<Файл>/<Компилировать выбранные блоки>) открывает диалоговую панель, в которой Вы выбираете нужные для компиляции блоки.

6. Диалоговое окно «Errors and Warnings (Ошибки и предупреждения)» показывает все ошибки и предупреждения, найденные в процессе компиляции. Исправьте все ошибки, выданные компилятором, а затем повторит процедуру, описанную выше.

## Таблица символов

Все папки «S7 program» имеют свою таблицу символов, которая содержит в себе описание исполняемых блоков программы. Вы можете открыть таблицу символов из SIMATIC Manager с помощью двойного щелчка на папке «Symbols».

Вы можете также открыть таблицу символов в LAD/STL/FBD редакторе с помощью команды меню <Options>/<Symbol Table>.

Когда Вы открываете символьную таблицу, появляется новое окно. Таблица содержит столбцы для символики (symbol), адреса (address), типа данных (data type) и комментария (comment). Каждое символьное имя занимает одну строку в таблице. Пустая строка автоматически добавляется в конце таблицы для ввода нового символа.

Недействительные символы обозначаются в этом столбце следующим:

= Символьное имя или абсолютный адрес идентичны другой строке данной таблицы символов.

X Неполный ввод (Отсутствует символьное имя и / или абсолютный адрес).

После создания символьной таблицы, ее могут использовать другие инструменты STEP 7 (такие как редактор LAD/STL/FBD, HW-Config и Monitor/Modify Variables).

## Сохранение проекта

Для сохранения открытого проекта можно воспользоваться меню <File>/<Save As…>, либо сочетанием клавиш «Ctrl + S». В появившемся окне нажать «Browse…» и выбрать каталог для сохранения проекта.

## Соединение с ПЛК

### Подключение программатора к CPU

Если Вы имеете USB-адаптер (6ES7972-0CB20-0XA0), то в качестве устройства программирования может использоваться любой персональный компьютер или ноутбук, который удовлетворяет требованиям к установке пакета STEP7 MS Windows ХР PRO, Vista Ultimate/Business, 7 32/64-Bit. USB- адаптер поддерживает протоколы MPI (S7-300/400), PPI (S7-200)n PROFIBUS-DP. Максимальная скорость передачи 1.5Mbps.

Для связи с S7-CPU Вы можете также использовать специальные коммуникационные модули SIMATIC NET, например. СР 5512 (PC-CARD) или СР 5611 (PCI-CARD).

### Проверка актуальности проекта в CPU

Для проверки актуальности проекта, загруженного на PLC, необходимо выполнить сравнение программных блоков.

Для этого выполнение следующие действия:

1. Переведите режим работы SIMATIC Manager в Offline (<View>/<Offline>).
2. Нажмите правой кнопкой мыши по папке блоков в обозревателе проекта и в появившемся контекстном меню выберите <Compare Blocks...>.
3. В появившемся окне сравнения выберите тип сравнения ONLINE/Offline и нажмите <Compare>.
4. После сравнения будет выведен список Offline блоков, отличающихся от ONLINE блоков.

### Тестирование сигналов ввода вывода из утилиты HW Config

Для этого выполните следующие действия:

1. Создать новый проект (<File>/<New...>) или открыть существующий (<File>/<Open...>).

2. Для нового проекта выделить имя проекта и выполнить функцию <PLC>/<Upload Station to PG...>.

3. Выделить объект с именем станции и выполнить функцию Open Object (контекстное меню) или дважды кликнуть мышью на объекте Hardware.

4. В окне утилиты HW Config вьщелить модуль ввода-вывода и Hardware выполнить функцию Monitor/Modify (контекстное меню).

5. В появившемся окне Monitor/Modify провести тестирование сигналов.

## Резервное копирование

Для выполнения работ по созданию резервной копии необходимо запустить SIMATIC Manager и открыть проект.

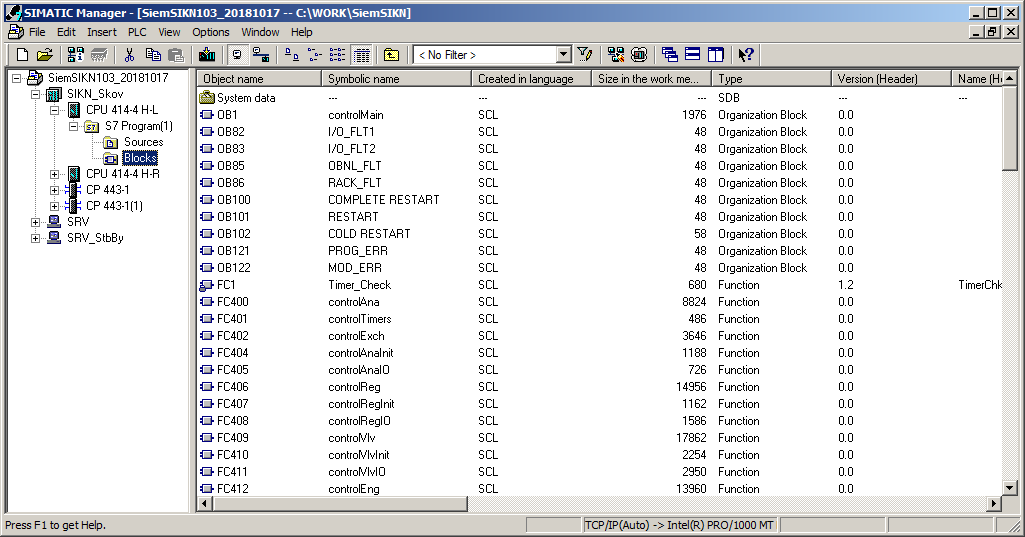


Рисунок 3.1 – Главное окно программы

Нажимаете правой клавишей мыши на проект и выбираете пункт меню, как на рисунке 3.2.

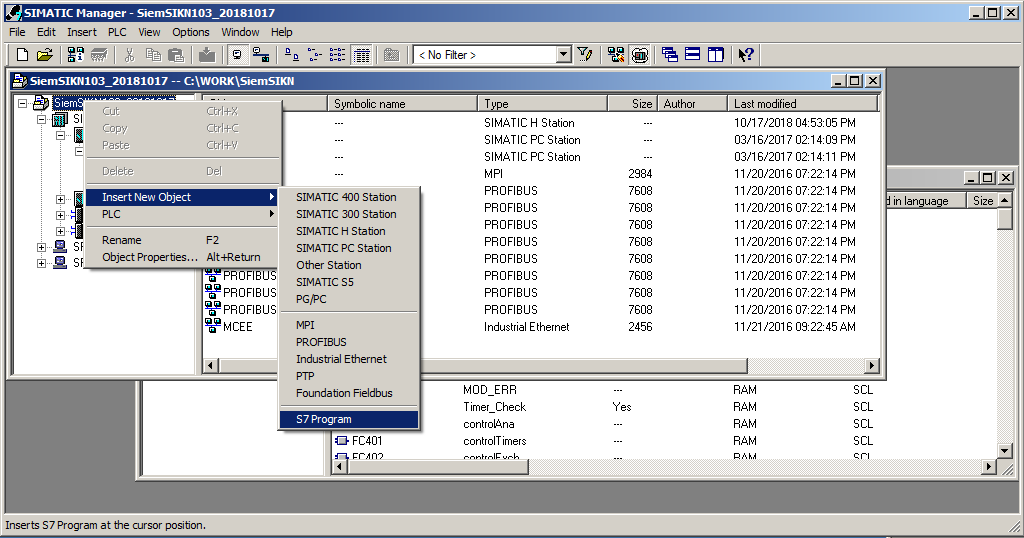


Рисунок 3.2 – Главное окно программы

Необходимо ввести название папки, например, текущую дату/время.

Переходим в режим подключения к ПЛК. Нажимаете View -> Online. Откроется окно, как на рисунке 3.3.

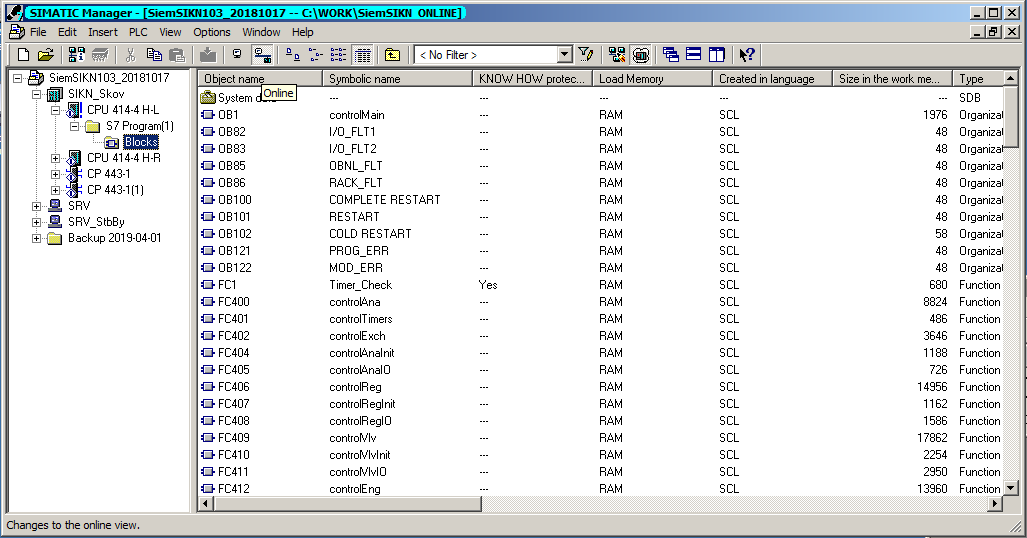


Рисунок 3.3 – Главное окно программы

Нажимаете правой клавишей мыши по папке Block и выбираете пункт меню Copy.

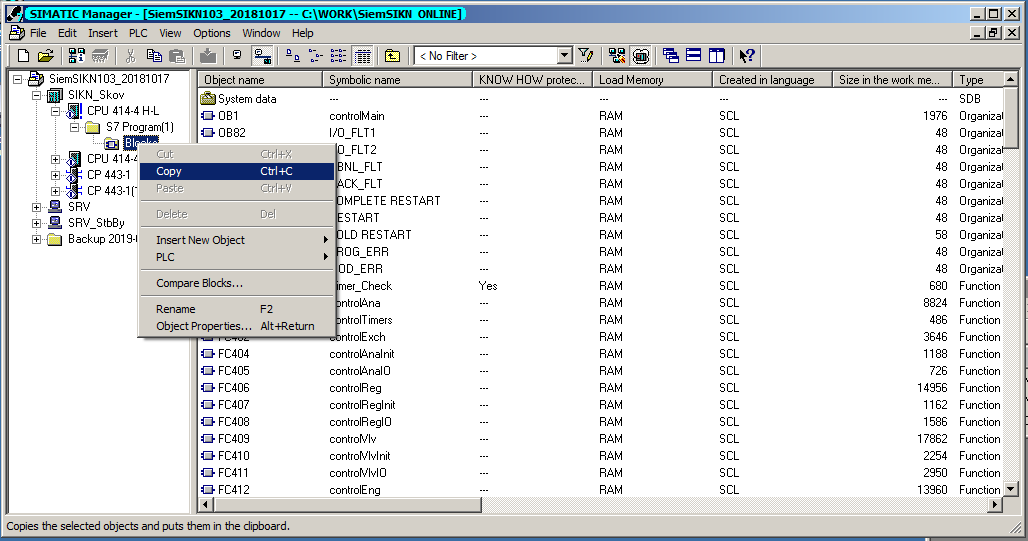


Рисунок 3.4 – Главное окно программы

После нажимаете на созданную ранее папку правой клавишей мыши и выбираете пункт меню Paste.

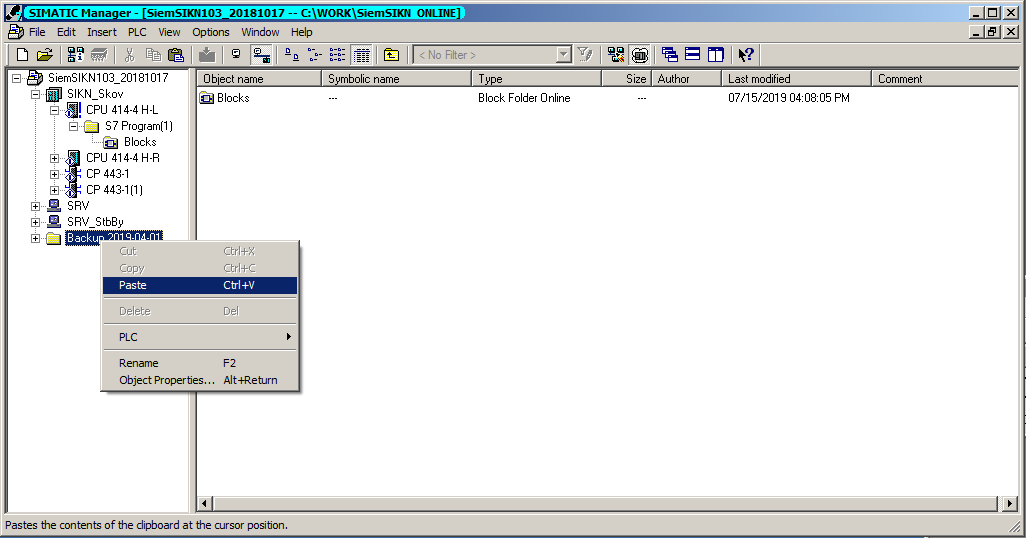


Рисунок 3.5 – Главное окно программы

Подтверждаете копирование.

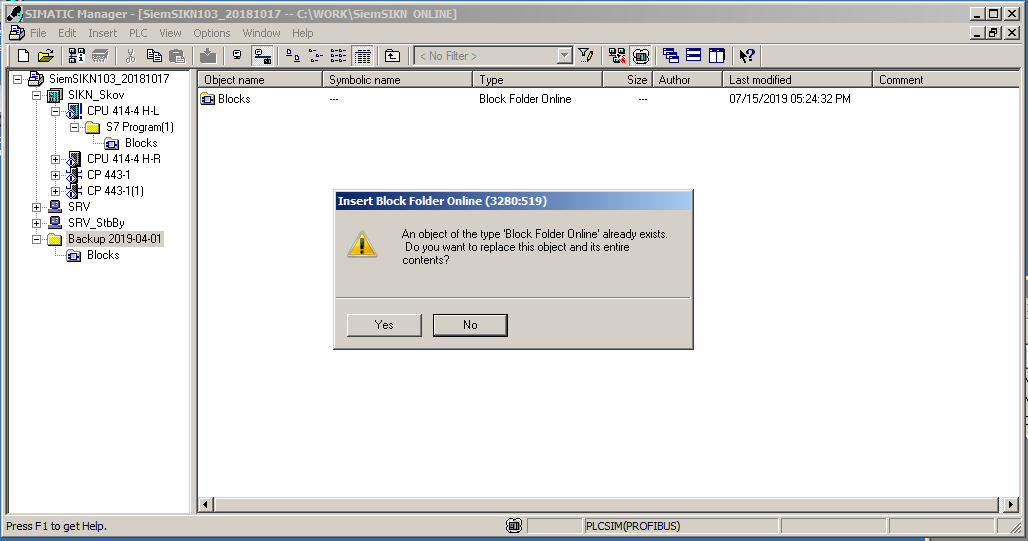


Рисунок 3.6 – Главное окно программы

Нажимаете на папку Block. При появлении окна (рисунок 3.7) нажимаете ОК, внизу окна на кнопку View, и снова ОК.

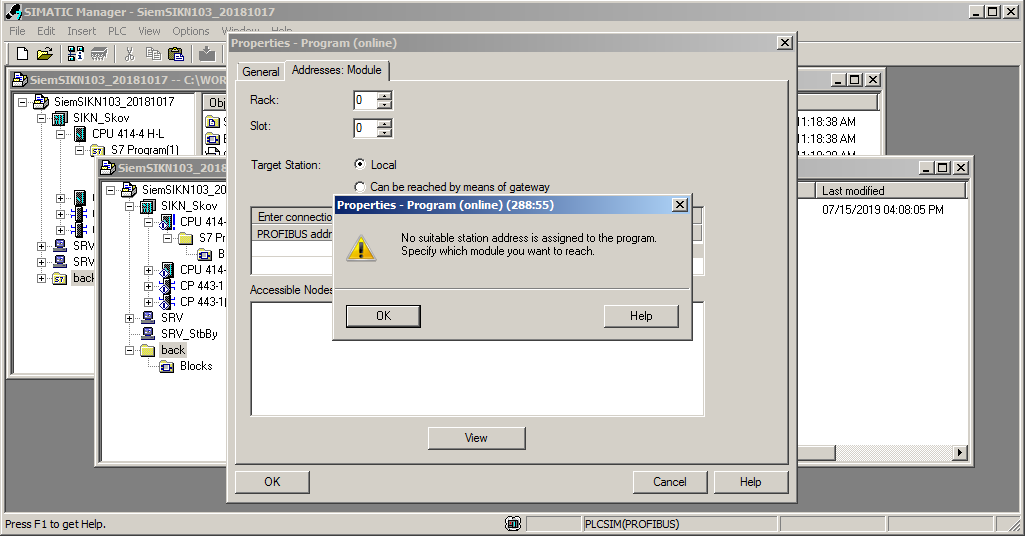


Рисунок 3.7 – Главное окно программы

Нажимаете правой клавишей мыши и выбираете пункт меню Upload to PG.

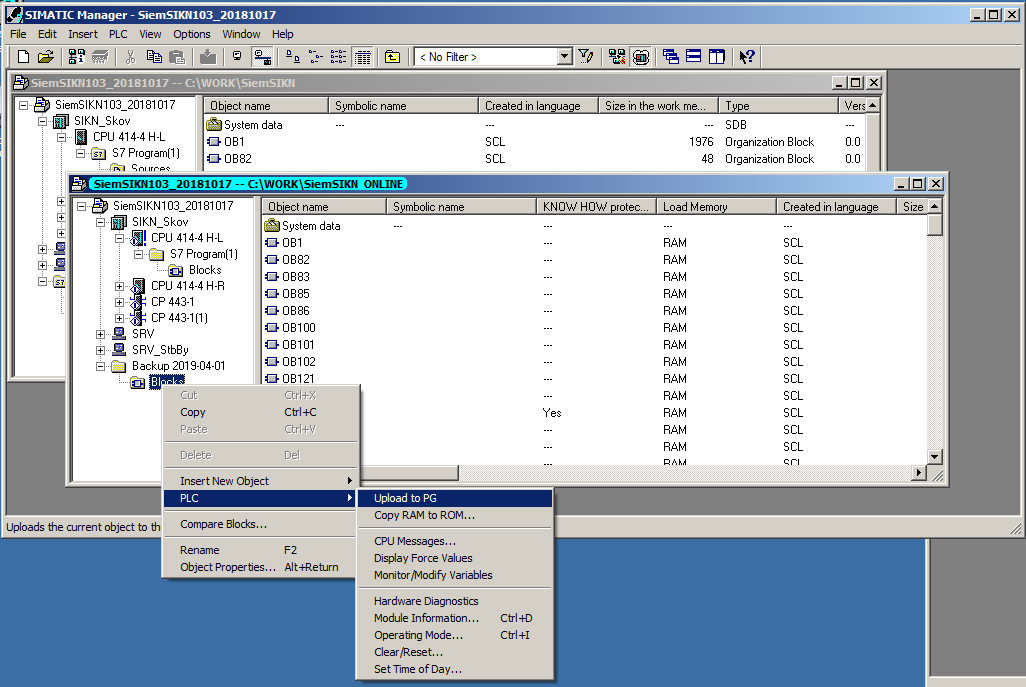


Рисунок 3.8 – Главное окно программы

При появлении окна (рисунок 3.9) нажать кнопку «All».

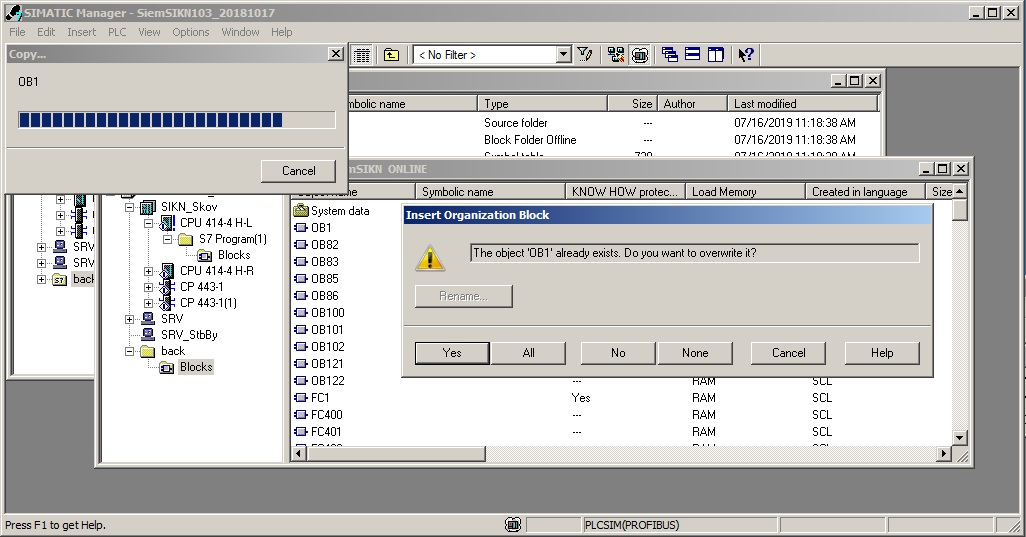


Рисунок 3.9 – Главное окно программы

После этого полная резервная копия программы исполнительных блоков с сохраненными параметрами (переменными, коэффициентами и уставками) готова.

## Восстановление из резервной копии

Для выполнения работ по восстановлению из резервной копии необходимо запустить SIMATIC Manager и открыть проект.

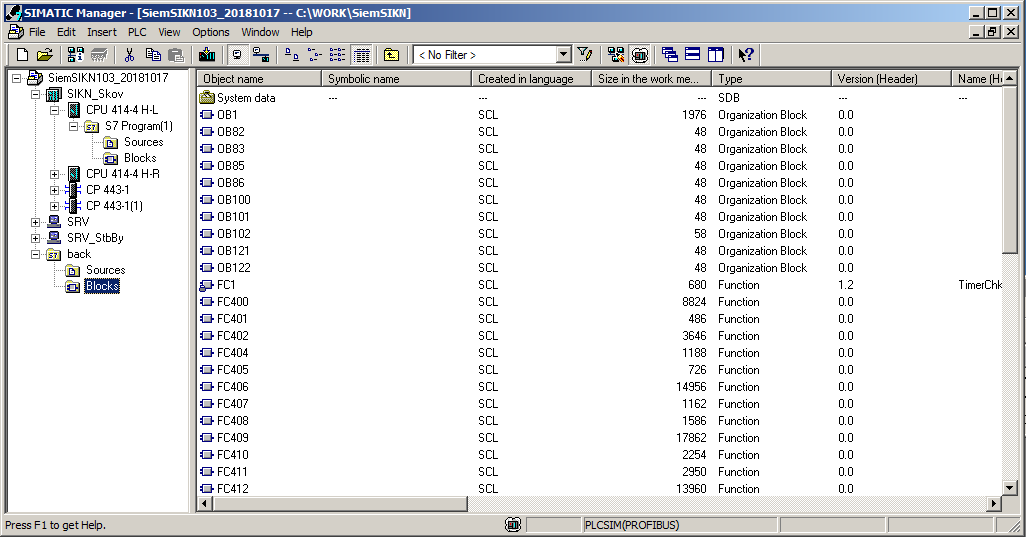


Рисунок 3.10 – Главное окно программы

Выбираете папку Block, нажимаете правой клавишей мыши и выбираете «Download», как на рисунке 3.11.

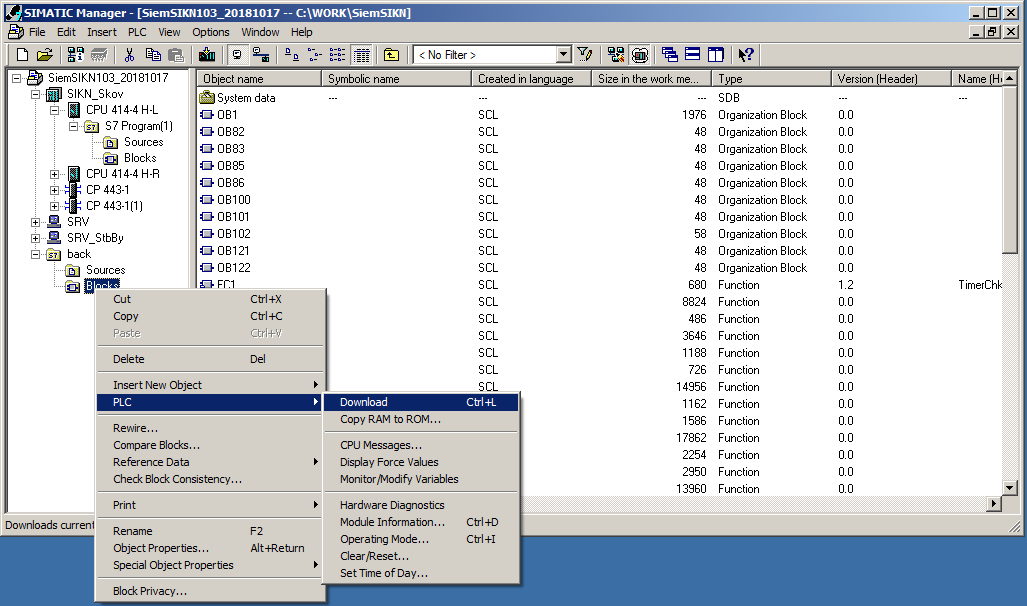


Рисунок 3.11 – Главное окно программы

При необходимости восстановить только функции выбрать конкретную функцию и загрузить, как на рисунке 3.12.

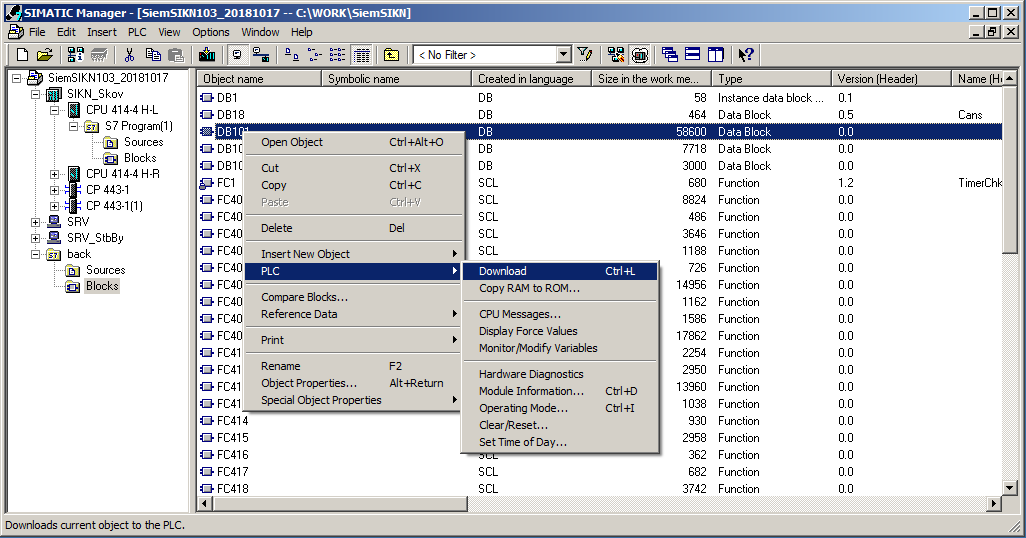


Рисунок 3.12 – Главное окно программы

При появлении окна, как на рисунке 3.13, нажать кнопку «All».

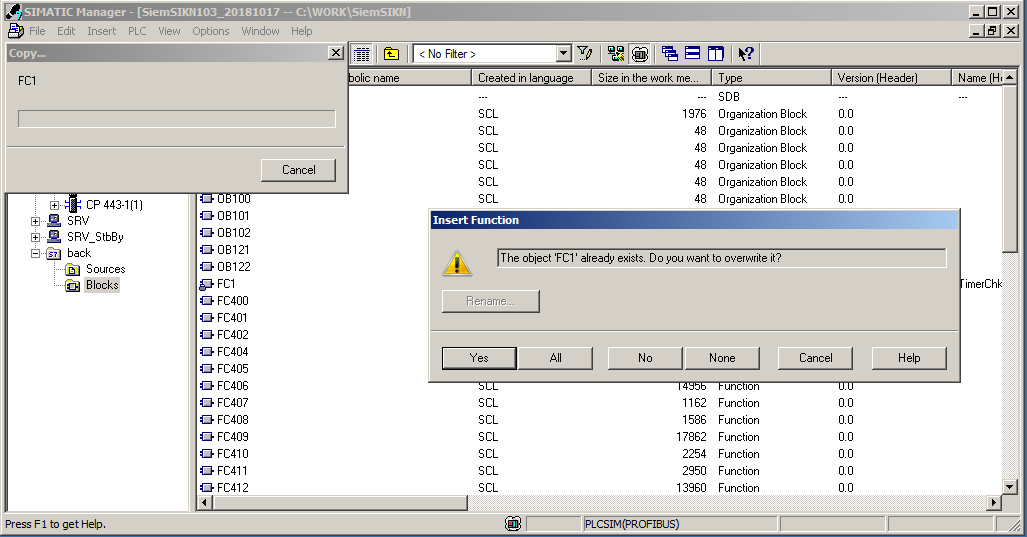


Рисунок 3.13 – Главное окно программы

При появлении сообщения, как на рисунке 3.14, нажать на кнопку «No».

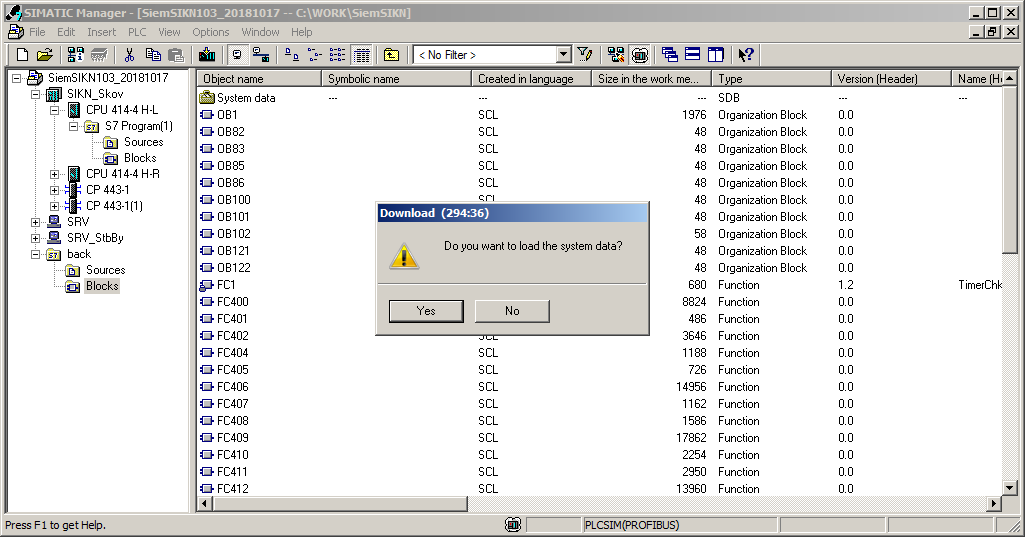


Рисунок 3.14 – Главное окно программы

# Введение в программирование в среду разработки SIMATIC Manager

## Общие сведения

Программа представляет собой совокупность структурных единиц (POU), определение которых приведено в стандарте МЭК 61131. Программа имеет прямой доступ ко всех глобальным переменным, функциям и функциональным блокам.

Программы могут быть созданы в среде SIMATIC Manager с использованием различных языков программирования. В рамках одного проекта допускается применение нескольких языков программирования.

Перечень поддерживаемых языков программирования приведен в таблице 4.1.

Таблица 4.1 – Перечень поддерживаемых языков программирования

|  |  |  |
| --- | --- | --- |
| Язык программирования | МЭК 61131 | Примечания |
| Язык лестничных диаграмм (LD) | Да | Графический |
| Язык диаграмм функциональных блоков (FBD) | Да | Графический |
| Язык непрерывных функциональных диаграмм (CFC) | Нет | Графический |
| Язык последовательных функциональных диаграмм (SFC) | Да | Графический и текстовый |
| Язык списка инструкций (IL) | Да | Текстовый |
| Язык структурированного текста (ST) | Да | Текстовый |
| ANSI C и C++ | Нет | Текстовый |

В среде SIMATIC Manager все текстовые языки программирования используют один и тот же редактор. Таким образом, используются одинаковые диагностические средства с одним и тем же набором функций. Такое единообразие позволяет упростить рабочий процесс и повысить производительность.

Функциональные блоки из стандартных библиотек Siemens могут вызываться из любого языка программирования и использоваться в любом языке программирования.

## Инициализация и циклические части программы

Каждая программа (задача) состоит из трех частей:

1) инициализация задачи;

2) циклическая часть задачи;

3) завершение задачи.

Перед запуском циклической задачи выполняется программа инициализации. Программа инициализации может содержать программный код, определяющий значения переменных.

Циклическая часть программы начинается после завершения инициализации задачи.

Переменные, которым присвоены значения, сохраняют их до тех пор, пока не получат новые значения или система не будет перезагружена.

Программа выхода вызывается при удалении задачи. Если при инициализации или выполнении циклической программы были использованы определенные ресурсы (память, интерфейсы), то они должны быть надлежащим образом освобождены.

## Переменные и типы данных

Переменные представляют собой участки памяти, предназначенные для хранения данных определенного приложения. Программирование упрощается за счет использования символических имен переменных.

Типы данных описывают такие свойства переменной, как например, диапазон или точность числа, содержащегося в переменной, или какие операции могут выполняться с ней.

Типы данных определяют:

• Тип и интерпретацию элементов данных,

• Диапазон значений для элементов данных,

• Множество операций, которые могут выполняться с адресом типа данных

• Способ записи элементов данных

*Элементарные типы данных*

Элементарные типы данных определяют структуру элементов данных, которая не может быть разделена на меньшие части. Они соответствуют стандарту DIN EN 1131-3. Элементарные типы данных описывают области памяти фиксированной длины и включают бит, целые, действительные, период времени, время дня и символьные величины. В SCL представлены следующие элементарные типы данных:

|  |  |  |
| --- | --- | --- |
| **Группа** | **Тип данных** | **Объяснение** |
| Битовый типы данных | BOOL  BYTE  WORD  DWORD | Элементы данных этих типов занимают 1 бит, 8 бит, 16 или 32 бита |
| Символьный тип | CHAR | Элементы данных этого типа занимают 1 символ в наборе символов ASCII |
| Численные типы | INT  DINT  REAL | Элементы данных этих типов доступны для обработки числовых величин. |
| Временные  типы | TIME  DATE  TIME\_OF\_DAY  S5TIME | Элементы данных этих типов представляют временные величины данных в соответствии с STEP 7. |

*Сложные типы данных*

SCL поддерживает следующие виды сложных типов:

|  |  |
| --- | --- |
| **Тип данных** | **Объяснение** |
| DATE\_AND\_TIME DT | Определяет область из 64 бит (8 байт). Этот тип данных содержит дату и время (в двоично-десятичных кодах). Это встроенный тип данных в SCL. |
| STRING | Определяет область для символьной строки, содержащей до 254 символов (тип данных CHAR). |
| ARRAY | Определяет массив, состоящий из элементов данных одного типа (элементарных или сложных). |
| STRUCT | Определяет группу данных любых типов, простых или сложных, в любых комбинациях. |

*Определенный пользователем тип данных*

Вы можете создавать свои типы данных, путем их декларации. Каждый из них имеет уникальное имя и может использоваться в программе многократно. Как только тип определен, этот тип данных может использоваться для создания нескольких блоков данных с одинаковой структурой.

Параметрируемый тип - это специальный тип данных для таймеров, счетчиков и блоков, которые используются как формальные параметры.

|  |  |
| --- | --- |
| **Тип данных** | **Объяснение** |
| TIMER | Используется для объявления таймеров как параметров. |
| COUNTER | Используется для объявления счетчиков как параметров. |
| BLOCK XX | Используется для объявления FC, FB, DB и SDB как параметров. |
| ANY | Используется как параметр для доступа к областям памяти с данными определенного типа. |
| POINTER | Используется как параметр для доступа к области памяти. |

*Тип данных ANY*

В SCL Вы можете использовать переменные типа ANY как формальный параметр блока. Вы также можете создать временные переменные этого типа и использовать их в назначении величин.

## Символьная адресация переменных

Символьная адресация (symbolic addressing) использует имена (символы) вместо абсолютных адресов. Вы сами можете выбирать эти имена. Такое имя должно начинаться с буквы и может содержать до 24 символов. В STL не разрешено использовать ключевые слова в качестве имен (символов). Для того, чтобы использовать ключевые слова в качестве имен (символов) в SCL, вставьте символ решетки «#» перед таким именем. При присвоении имен входам учитывается регистр написания символа (имеет значение, какой регистр применяется: верхний или нижний). Для имен выходов редактор использует регистр и нотацию (форму записи), которые были применены при объявлении (declaration) символа.

При символьной адресации абсолютному адресу должно быть назначено имя (символ).Символы различаются по месту назначения: глобальные символы действительны во всей программе, тогда как локальные символы действительны только в блоке, в разделе объявления переменных которого они описаны.

*Глобальные символы*

Вы можете назначить имена в таблице символов (symbol table) следующим объектам:

• Блоки данных и кодовые блоки

• Входы, выходы, периферийные входы и периферийные выходы

• Меркеры, таймеры и счетчики

• Пользовательские типы данных

• Таблицы переменных

Глобальный символ может также содержать пробелы, специальные символы и национальные символы, такие как умляут. Исключения составляют символы 00hex, FFhex и кавычки ("). При использовании в программе имен со специальными символами Вы должны заключать имена в кавычки. В скомпилированном блоке программный редактор всегда отображает все глобальные символы в кавычках.

Вы можете использовать глобальный символ во всей программе; каждый такой символ должен быть уникальным (однозначно принадлежать одному адресу) в этой программе.

*Локальные символы*

Имена локальных данных определяются в разделе объявления переменных соответствующего блока. Эти имена могут содержать только буквы, цифры и знак подчеркивания.

Локальные символы являются действующими только внутри блока, в котором они описаны. Такие же символы (такие же имена переменных) могут быть применены в ином контексте (для обозначения совершенно иных объектов) в другом блоке. Редактор отображает локальные символы (имена) с впереди стоящим символом «#». Если редактор не может отличить локальный символ от адреса Вы должны вводить этот символ с впереди стоящим символом «#». Локальные символы доступны только в базе данных программатора PG (в автономном [offline] каталоге Blocks [Блоки]). Если эта информация отсутствует при декомпиляции, то редактор вставляет символы замены (substitute symbol).

*Использование символьных имен*

Если Вы используете символьные имена во время инкрементного программирования, то эти имена должны уже к этому времени быть присвоены абсолютным адресам. Вы можете ввести новые символические имена в таблицу символов во время инкрементного программирования и можете в дальнейшем использовать в программе.

Если вы программируете исходный текстовый файл программы, то полностью закончить процесс назначения символических имен абсолютным адресам необходимо лишь к моменту компиляции программы.

В случае использования массивов доступ к отдельным элементам массивов обеспечивается использованием имени массива с индексом, например, имя MSERIES[1] принадлежит первому элементу массива MSERIES. В случае программирования на STL индекс должен быть константой (INT). В случае программирования на SCL индекс может быть как целой переменной (INT), так и целым выражением (INT).

В структурах каждый элемент имени («subname») отделяется от остальных элементов десятичной точкой, например, FRAME.HEADER.CNUM.

Компоненты пользовательских типов данных адресуются точно также как и компоненты структур.

*Адресация данных*

Символьная адресация данных предполагает использование полного адреса, включая блока данных. Например, блок данных с символьным адресом MVALUES содержит переменные MVAL1, MTIME. Эти переменные могут быть адресованы следующим образом:

"MVALUES".MVAL1

"MVALUES".MTIME

# ПРИЛОЖЕНИЕ А

**Общий вид окон SIMATIC Manager**



Рисунок А1.1 – Начальная страница

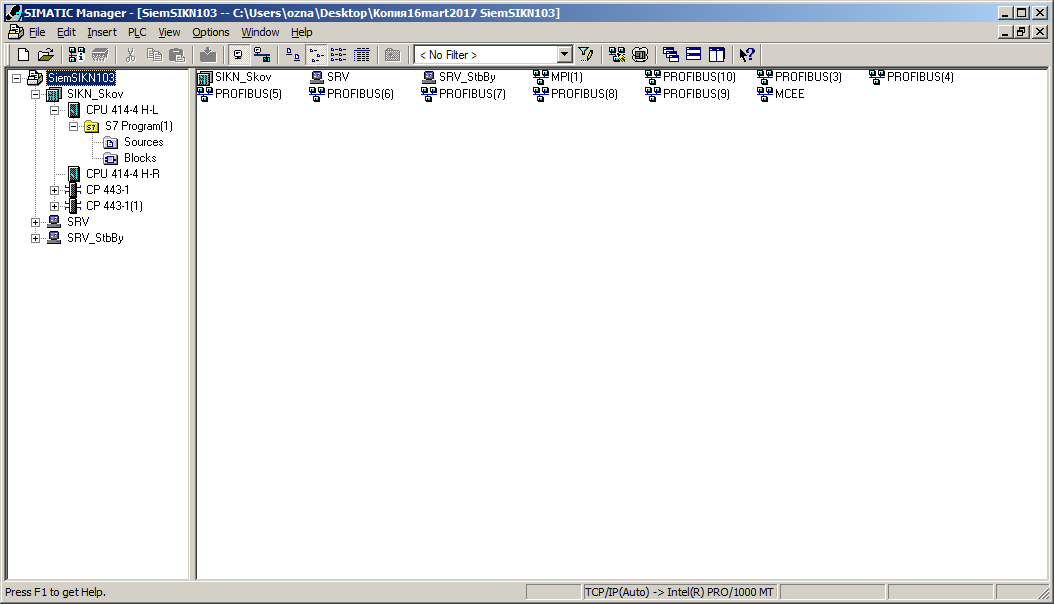


Рисунок А1.2 – Рабочие области

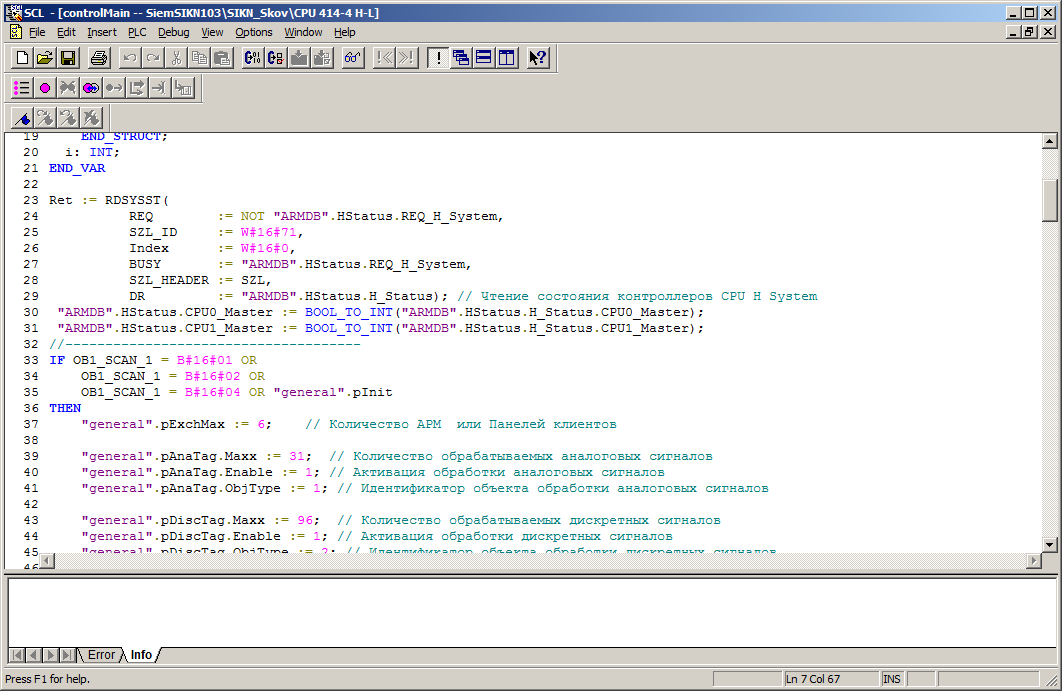


Рисунок А1.3 – Конфигурация программного обеспечения

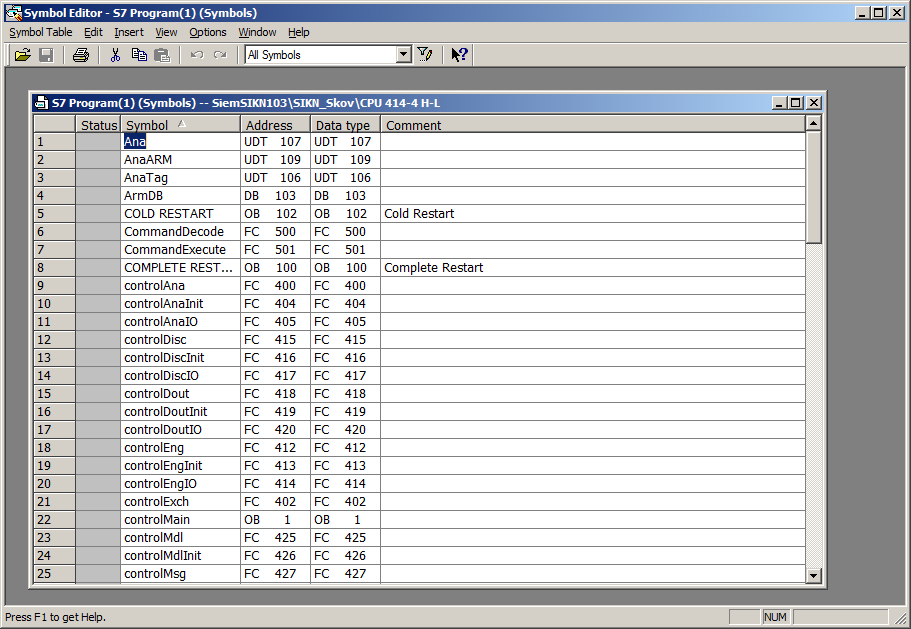


Рисунок А1.4 – Редактор «I/O Mapping»

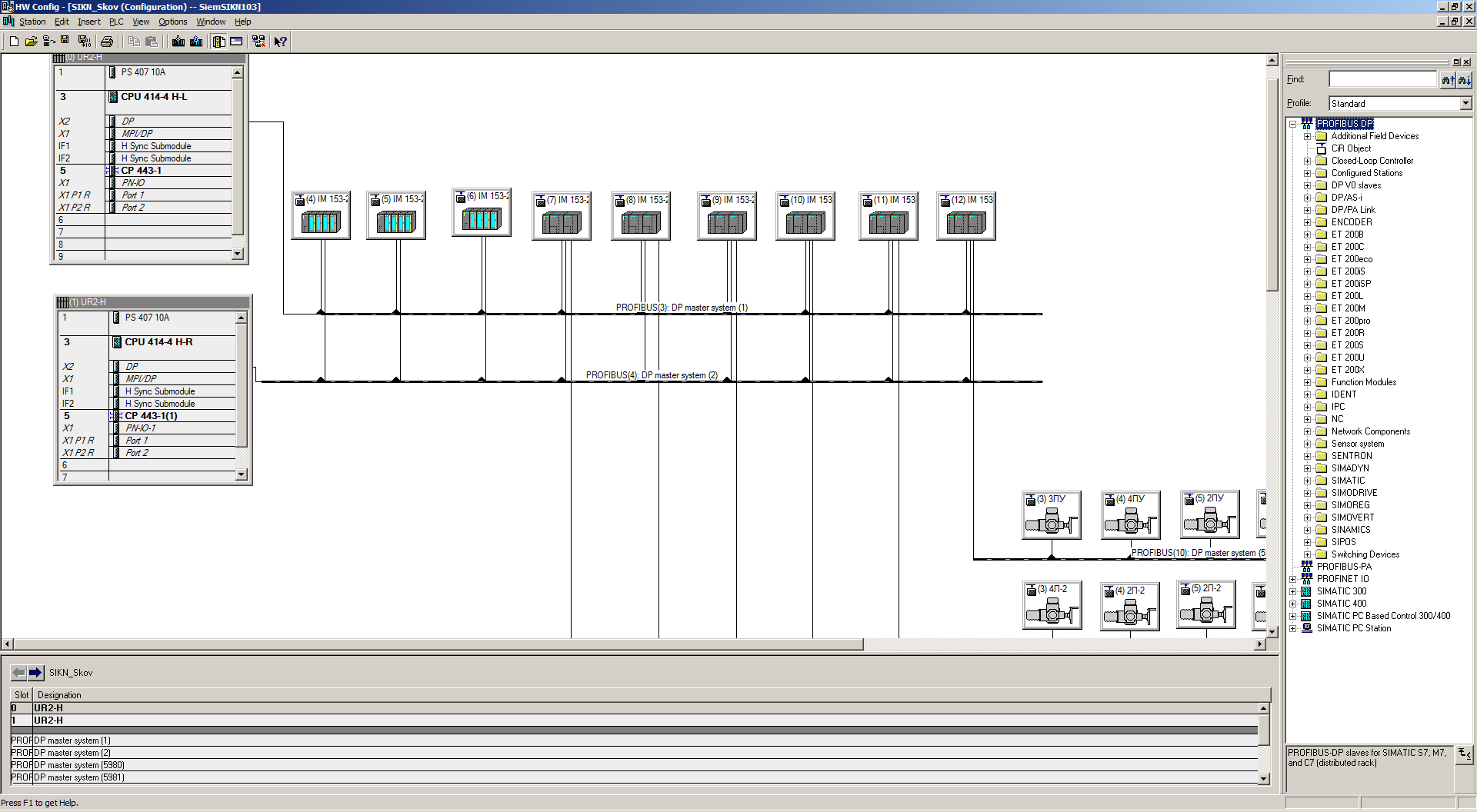


Рисунок А1.5 – Конфигурация аппаратного обеспечения

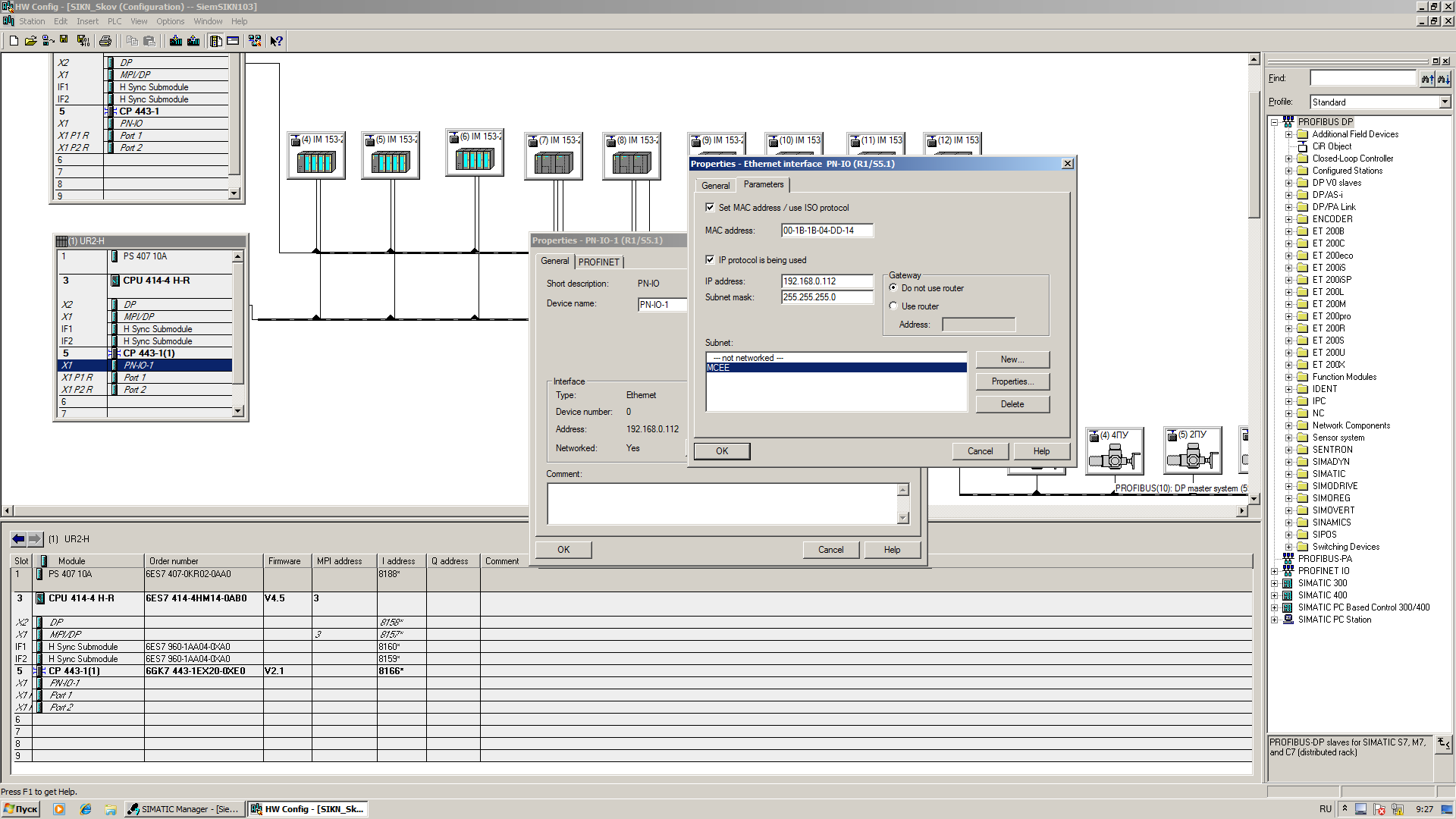


Рисунок А1.6 – Настройка IP–адреса сетевого интерфейса контроллера

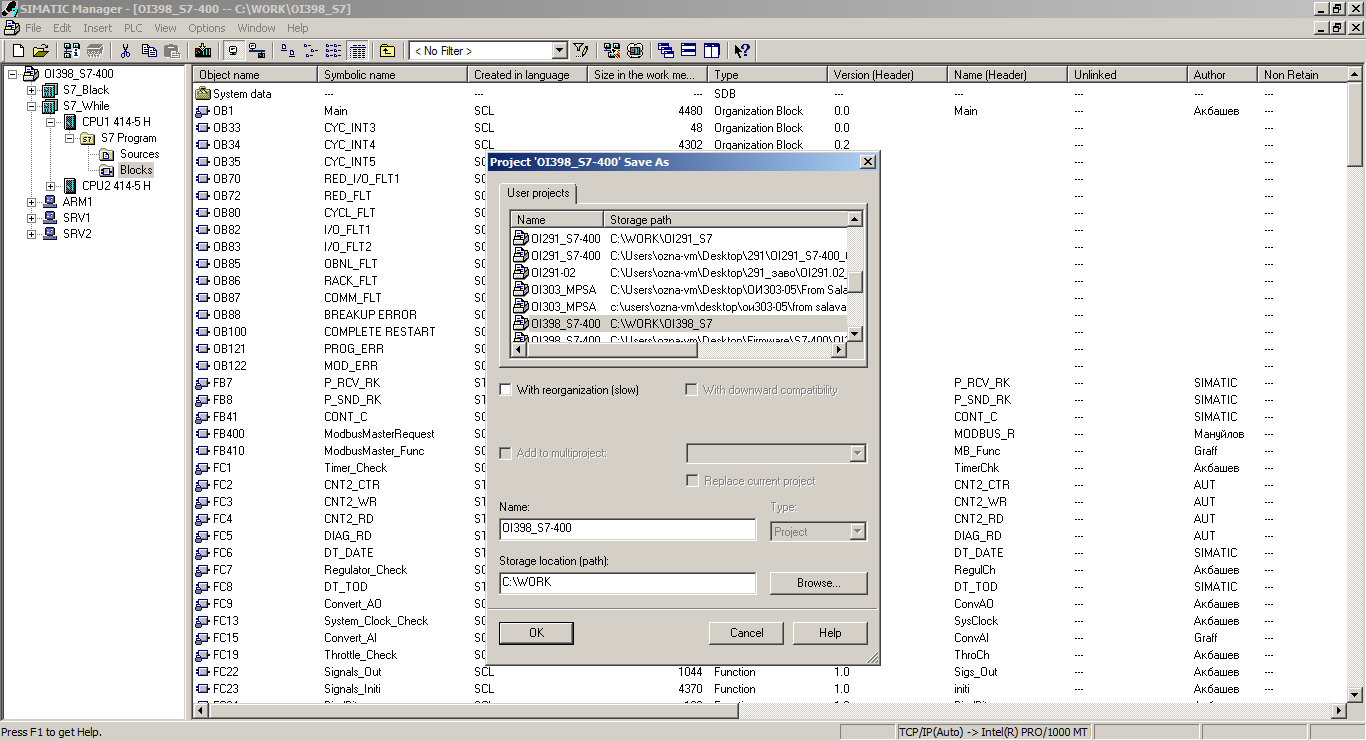


Рисунок А1.7 – Меню сохранения текущего проекта

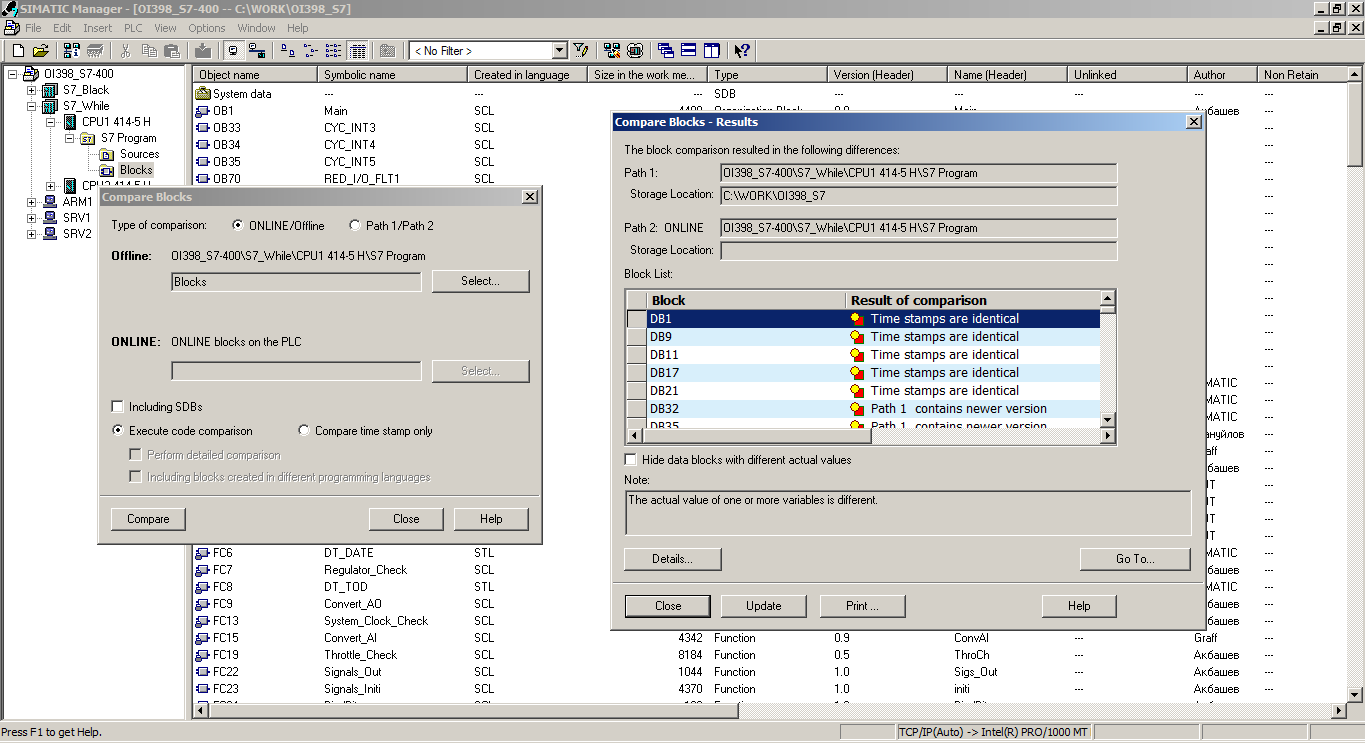


Рисунок А1.8 – Меню сравнения блоков и меню результата сравнения

# Лист регистрации изменений

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Изм. | Номера листов | | | | Всего листов в документе | № документа | Подпись | Дата |
| Измененных | Замененных | Новых | Аннулированных |
| 1 |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |  |
| 17 |  |  |  |  |  |  |  |  |
| 18 |  |  |  |  |  |  |  |  |
| 19 |  |  |  |  |  |  |  |  |
| 20 |  |  |  |  |  |  |  |  |
| 21 |  |  |  |  |  |  |  |  |
| 22 |  |  |  |  |  |  |  |  |
| 23 |  |  |  |  |  |  |  |  |
| 24 |  |  |  |  |  |  |  |  |
| 25 |  |  |  |  |  |  |  |  |
| 26 |  |  |  |  |  |  |  |  |
| 27 |  |  |  |  |  |  |  |  |
| 28 |  |  |  |  |  |  |  |  |
| 29 |  |  |  |  |  |  |  |  |
| 30 |  |  |  |  |  |  |  |  |
| 31 |  |  |  |  |  |  |  |  |
| 32 |  |  |  |  |  |  |  |  |
| 33 |  |  |  |  |  |  |  |  |
| 34 |  |  |  |  |  |  |  |  |