



**Руководство пользователя:
CoDeSys V3, установка и первый запуск**

Редакция 3.0

СОДЕРЖАНИЕ

1	ДОБРО ПОЖАЛОВАТЬ	3
2	СИСТЕМНЫЕ ТРЕБОВАНИЯ	3
3	УСТАНОВКА	3
4	ЗАПУСК	3
5	ПОМОЩЬ	3
6	СОЗДАНИЕ И ЗАПУСК ПРОЕКТА	4
6.1	Запуск CoDeSys и создание проекта	4
6.2	Разработка программы	6
6.3	Настройка ресурсов CoDeSys для запуска и управления приложением в ПЛК	9
6.4	Запуск и мониторинг приложения в ПЛК	12
6.5	Отладка приложения	15
7	УДАЛЕНИЕ, НАСТРОЙКА, ВОССТАНОВЛЕНИЕ	16
8	ИСТОРИЯ ИЗМЕНЕНИЙ ДОКУМЕНТА	17

1 Добро пожаловать

Добро пожаловать в CoDeSys V3.x – систему разработки от компании 3S - Smart Software Solutions GmbH!

CoDeSys – это аппаратно-независимая система программирования ПЛК. Полностью отвечая требованиям МЭК 61131-3, она поддерживает все стандартные языки программирования, позволяет использовать процедуры на С и поддерживает объектно-ориентированное программирование. В сочетании с системой исполнения CoDeSys SP возможна работа с несколькими устройствами и приложениями. Благодаря структурному разбиению на компоненты, вы можете сами конфигурировать и расширять пользовательский интерфейс.

Перед установкой продукта ознакомьтесь с системными требованиями.

2 Системные требования

Необходимо:

- ОС: Windows 2000, Windows XP
- ОЗУ: 512MB
- Диск: 200MB
- Процессор: Pentium IV, Centrino > 1,8 GHz, Pentium M > 1,0GHz

Рекомендуется:

- ОС: Windows XP
- ОЗУ: 1024MB
- Диск: 1GB
- Процессор: Pentium IV, Centrino > 3,0 GHz, Pentium M > 1,5GHz

3 Установка

Запустите файл *Setup_CoDeSysV<Version>.exe*; далее следуйте указаниям мастера установки.

4 Запуск

Запустите CoDeSys из меню *Пуск*; путь по умолчанию Programs – 3S CoDeSys – CoDeSys – CoDeSys V<version>.

Также после установки на *Рабочем столе* появится иконка CoDeSys .

5 Помощь

Оперативная помощь доступна в меню 'Help'. Вы можете найти нужную информацию по оглавлению, а также использовать поиск по индексу или ключевым словам.

6 Создание и запуск проекта

Здесь вы узнаете, как создать простой проект, представляющий собой ПЛК-программу, как загрузить его через Gateway Server в ПЛК (целевое устройство), как его запустить и проверить. Система исполнения ПЛК, используемая в этом примере, включена в установочный пакет CoDeSys.

Пример написан на языке ST (Structured Text) и состоит из программы *PLC_PRG* и функционального блока *FB1*. *PLC_PRG* содержит счетчик в переменной *ivar* и включает вызов экземпляра функционального блока *FB1*. *FB1* на входе "in" получает некоторое числовое значение от *PLC_PRG*, прибавляет к нему "2" и возвращает результат на выходе *out*; *out* будет считан в *PLC_PRG*.

(Имейте в виду, что приведенные ниже описания относятся к конфигурации пользовательского интерфейса CoDeSys заданной по умолчанию.)

6.1 Запуск CoDeSys и создание проекта

(1) Запуск CoDeSys

В меню **Пуск** выберете:

Programs > 3S CoDeSys > CoDeSys > CoDeSys V<version>

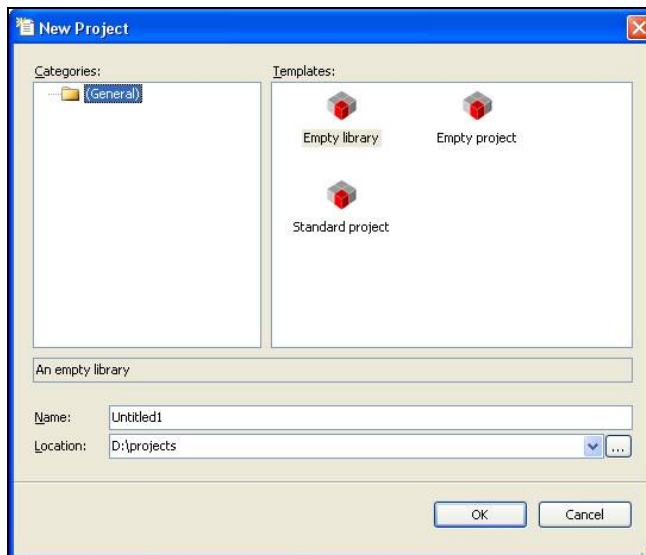


Также можно запустить CoDeSys через иконку **CoDeSys V <version>** на Рабочем столе.

Система программирования автоматически откроется со стандартным профилем, так что в данном примере его выбирать не надо. В дальнейшем, когда на компьютере будут установлены различные версии компонентов CoDeSys, вам будет необходимо выбрать нужный профиль системы при старте.

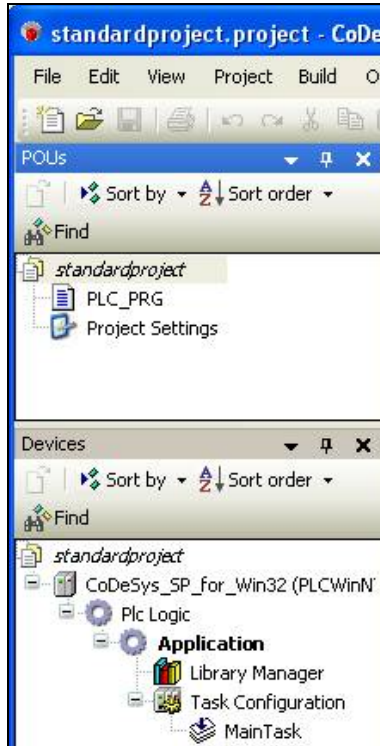
(2) Создание проекта

Для создания нового проекта выберете команду **New project** в меню **File**:



В окне **'Templates'** выберете **Standard project** и введите желаемое имя (**Name**) и директорию (**Location**) для сохранения проекта. Нажмите **OK**, чтобы создать новый проект.

Имя проекта (на рисунке `standardproject`) теперь появится в названии главного окна CoDeSys и в названии корневого узла в окнах **'POUs'** и **'Devices'**.



По умолчанию, в новом стандартном проекте доступны один программный компонент (POU), настройки проекта (Project Settings) и некоторые ресурсы:

В окне **'POUs'** (сверху) находится программа "PLC_PRG", которую нам предстоит написать на языке ST. В окне **'Devices'** (снизу) присутствует единственное устройство: "CoDeSys_SP_for_Win32 (PLCWinNT)". Оно содержит одно приложение (Application) с Конфигуратором задач (Task Configuration), определяющим главную задачу (MainTask), которая включает вызов PLC_PRG, и Менеджер библиотек (Library Manager). Он изначально содержит библиотеку "iostandard.library", необходимую для конфигурации входов/выходов.

При необходимости, вы можете переименовать устройство. Например, так: выберите устройство, нажмите **<Пробел>**, откройте поле редактирования и вместо "CoDeSys_SP_for_Win32" введите "MyPlc". Теперь устройство называется "MyPlc (PLCWinNT)".

(Пока не обращайте внимания на строчку "Plc Logic", расположенную под "MyPlc". Это всего лишь символический узел, указывающий на то, что наше устройство программируемое.)

6.2 Разработка программы

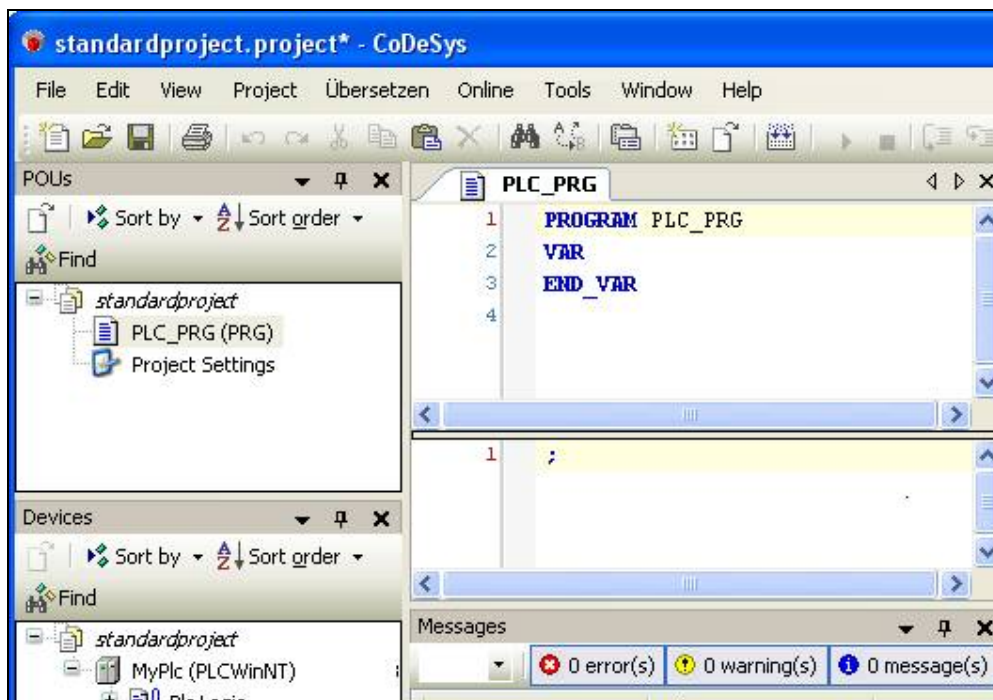
(3) Объявление переменных в PLC_PRG

Щелкните дважды мышкой по программному компоненту "PLC_PRG" в окне 'POUs'.

В центральной части экрана откроется окно **ST language editor** для 'PLC_PRG'. Редактор состоит из раздела объявлений (верхняя часть) и раздела кода (нижняя часть).

В разделе объявлений слева показаны номера строк, а также тип POU (в данном случае программа: "PROGRAM PLC_PRG") и ключевые слова "VAR" и "END_VAR" для объявления переменных.

По умолчанию текст программы состоит из единственного оператора ";":



В разделе объявлений поместите курсор после слова **VAR** и нажмите Ввод. В новой пустой строке объявите переменные *ivar* и *erg* типа INT и *fbinst* типа FB1:

```
PROGRAM PLC_PRG
VAR
    ivar: INT;
    fbinst: FB1;
    erg: INT;
END_VAR
```

В принципе, в CoDeSys не обязательно объявлять заранее все переменные, а можно сразу начать их использование в программе. В этом случае, сработает автоматическое определение переменных: 'Autodeclare', см. ниже (4).

(4) Реализация программного кода PLC_PRG

В разделе кода редактора PLC_PRG введите следующее:

```
ivar := ivar+1;           //счетчик
fbinst(in:=I1, out=>erg); //вызов экз. функц. блока FB1 с параметрами
erg := fbinst.out;       //запись результата в "erg"
```

Вместо шагов (3) и (4) можно использовать функцию Auto Declaration (автообъявление): просто введите строчку кода и нажмите <Ввод>. Для каждой еще не объявленной переменной будет открываться диалоговое окно 'Auto Declare', в котором вы можете задать необходимые параметры:

Поля 'Name', 'Scope' и 'Object' заполняются автоматически. Введите для переменной тип (Type) и комментарий (Comment), как описано в пункте (3). Обратите внимание, что данный комментарий не будет выведен в строке объявления после "//", как показано выше. Он будет сохранен в виде xml-описания, которое в дальнейшем может быть использовано при составлении документации проекта.

Закройте окно, нажав OK. Теперь переменная ivar появится в разделе объявлений нашего POU. Комментарий в данном случае будет выведен над переменной ivar, как показано здесь:

```
PLC_PRG | FB1 |
1 PROGRAM PLC_PRG
2 VAR
3   fbinst: FB1;
4   erg: INT;
5   /// <summary>
6   /// (* counter *)
7   /// </summary>
8   ivar: INT;
9 END_VAR
```

(5) Создание дополнительного POU (функционального блока FB1 на языке ST)

Мы включим в проект функциональный блок, который будет прибавлять "2" к входу "in". Результат будет записан на выход "out":

Выберете команду **Add object** в меню **Project**.

Выберете 'POU' в левой части окна 'Add Object'. Введите имя "FB1" для POU, установите флажок **Function Block** в разделе **Type**.

Выберете 'Structured Text (ST)' в поле **Implementation language**.

Подтвердите настройки объекта, нажав **Open**.

Откроется окно редактора для нового функционального блока FB1. Задайте здесь следующие переменные тем же методом, что и в PLC_PRG:

```
FUNCTION_BLOCK FB1
VAR_INPUT
    in:INT;
END_VAR
VAR_OUTPUT
    out:INT;
END_VAR
VAR
    ivar:INT:=2;
END_VAR
```



В разделе кода введите следующее:

```
out:=in+ivar;
```


6.3 Настройка ресурсов CoDeSys для запуска и управления приложением в ПЛК



(6) Запуск Gateway Server PLC

Gateway Server:

Gateway Server запускается автоматически вместе с операционной системой, как служебная программа. Убедитесь, что на панели задач есть красная иконка , которая указывает на то, что сервер запущен. Черная иконка  означает, что сервер отключен.

(Эта иконка является частью программы GatewaySysTray, которая служит для управления и контроля за работой Gateway. Она снабжена командами запуска и остановки, и, таким образом, позволяет пользователю приостанавливать или перезапускать сервер вручную. В будущем будет доступна функция Gateway Inspector. Меню также содержит команду 'Exit Gateway Control', которая просто отключает программу GatewaySysTray, в то время как Gateway может продолжать работу. Программа GatewaySysTray запускается автоматически вместе с Windows, однако, ее можно запустить и вручную через меню Programs.)

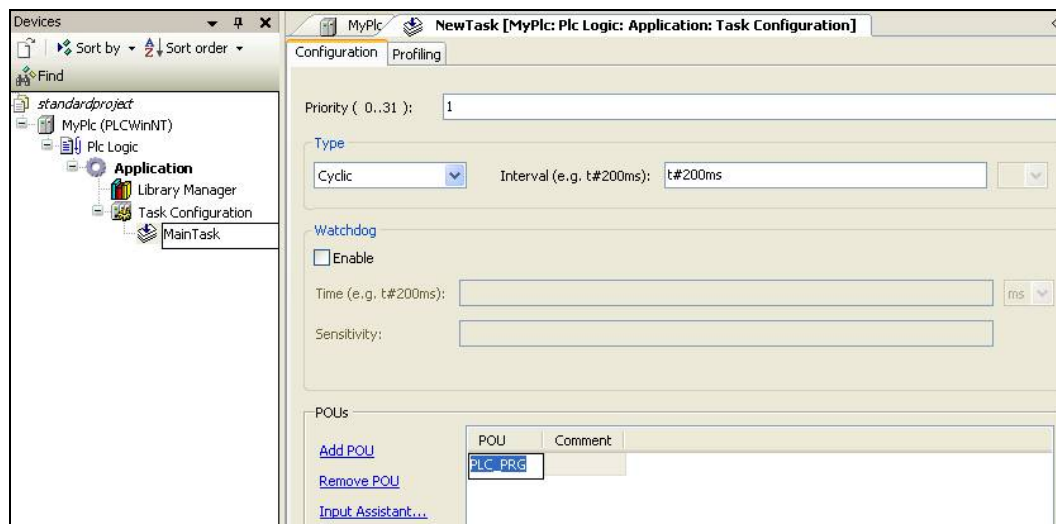
Запуск ПЛК:

ПЛК (CoDeSys SP) доступен, как служебная программа. На панели задач есть соответствующая иконка:  (ПЛК запущен) и  (ПЛК остановлен). Если позволяет система, программа будет автоматически запускаться при запуске системы. В противном случае программу надо запускать вручную командой 'Start PLC' из контекстного меню, открывающегося при щелчке правой кнопки мыши по иконке PLC.

(Эта иконка является частью программы GatewaySysTray, которая служит для управления и наблюдения за работой CoDeSysSP. Она снабжена командами запуска и остановки, и, таким образом, позволяет пользователю приостанавливать или перезапускать работу вручную. Меню также включает команду 'Exit PLC Control', которая просто отключает программу CoDeSysSPSysTray, в то время как PLCservice может продолжать работу. Программа GatewaySysTray запускается автоматически вместе с Windows, однако, ее можно запустить и вручную через меню Programs.)

(7) Задание активного приложения

Стандартный проект (Standardprojekt) по умолчанию содержит устройство "CoDeSys_SP_for_Win32 (PLCWinNT)", с которым связано приложение "Application" и конфигурация задач "Task Configuration":



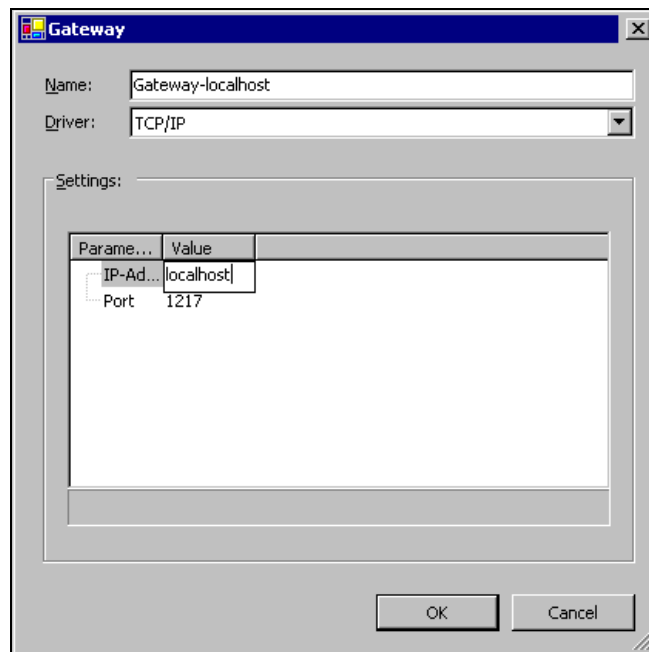
Нам необходимо сделать это приложение активным. Выберите 'Application' в окне 'Devices' и команду **Set Active Application** из контекстного меню.

(8) Настройка канала связи с ПЛК

Щелкните два раза на "MyPlc..." в окне 'devices'. Откроется диалог "MyPlc", содержащий окно "**Communication**". Здесь необходимо задать Gateway сервер, который будет использоваться системой программирования и настроить канал связи сервера с устройством. Выполните следующие шаги:

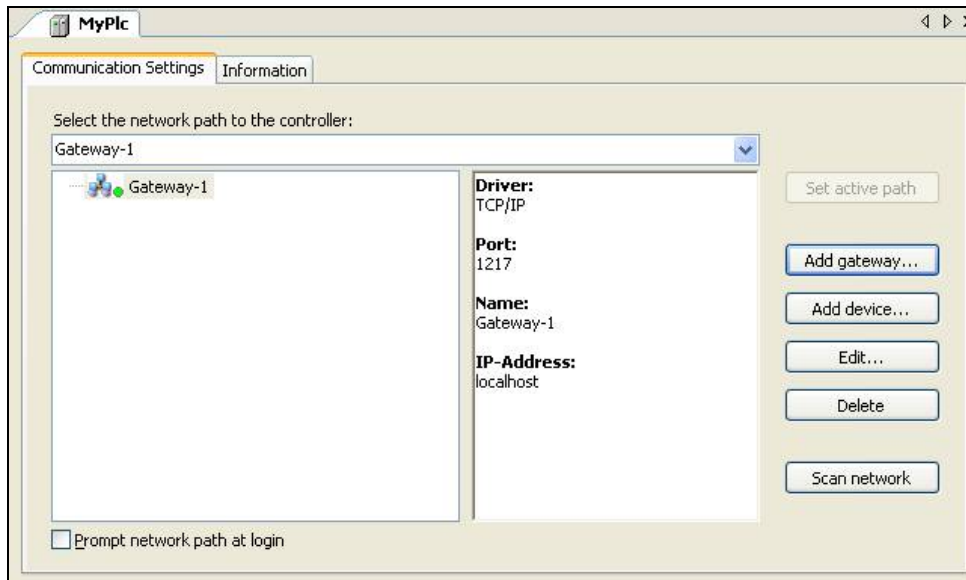
Если вы устанавливаете соединение в CoDeSys V3 в первый раз, то вам необходимо определить локальный Gateway Server. (Если вы уже определили сервер ранее, он будет указан в окне 'communication settings', как показано на следующей странице. В этом случае вы можете пропустить данный шаг и приступить к определению канала связи с устройством – см. следующую страницу).

Gateway сервер включен в установочный пакет CoDeSys. Нажмите **Add gateway**, чтобы открыть соответствующее диалоговое окно:



Введите желаемое символическое имя (**Name**) для Gateway, определите тип драйвера **Driver** "TCP/IP" и введите IP-адрес "localhost" (поле можно редактировать после двойного щелчка). Поле **Port** не изменяйте. Закройте окно, нажав **OK**.

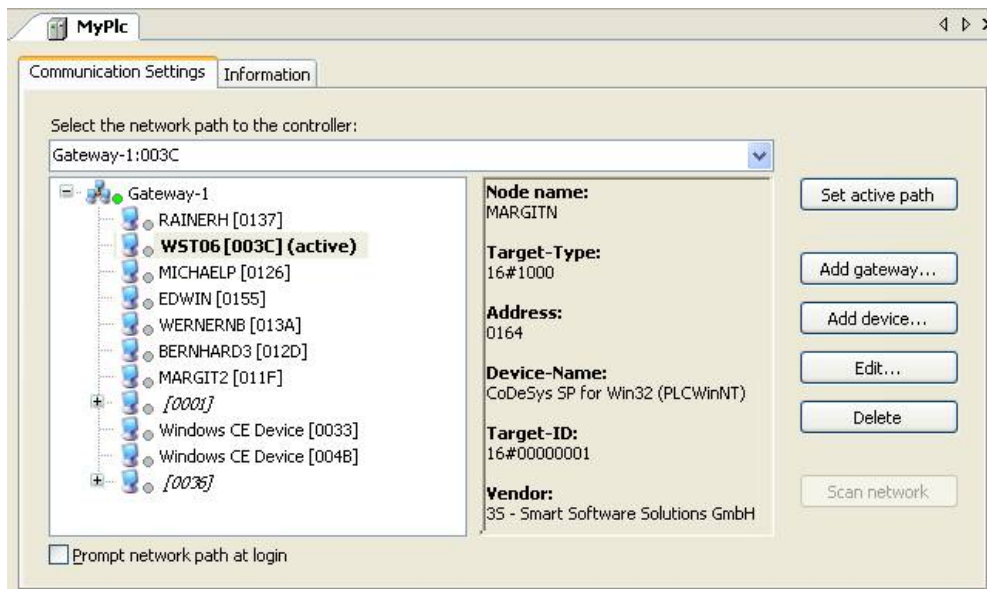
Название сервера будет помещено в левой части окна, а имя Gateway будет добавлено в список под надписью '**Select the network path to the controller**'. Если сервер работает, рядом будет гореть зеленый индикатор, в противном случае он будет красным:



Теперь определите канал связи Gateway сервера с устройством:

Нажмите **Scan network** для поиска доступных данному серверу устройств в локальной сети.

Как минимум вы найдете один ПЛК, установленный вместе с CoDeSys, он появится в списке под надписью Gateway. Вместо ПЛК "WST06 [003C]" вы увидите имя и адрес вашего компьютера, на котором работает устройство.



Теперь выберите ПЛК (устройство) и нажмите кнопку **Set active path**.

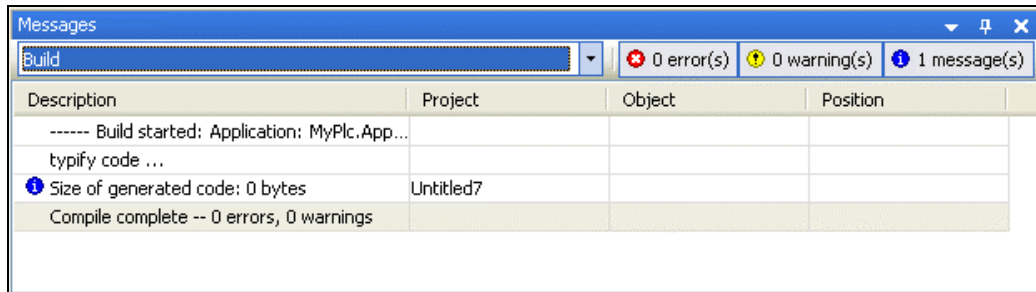
Таким образом вы активируете свой канал связи. Начиная с этого момента, все команды онлайн будут выполняться через этот канал. Таким образом можно иметь несколько каналов связи с разными устройствами в одном проекте.

Настройки будут приняты автоматически, теперь можно закрыть это окно.

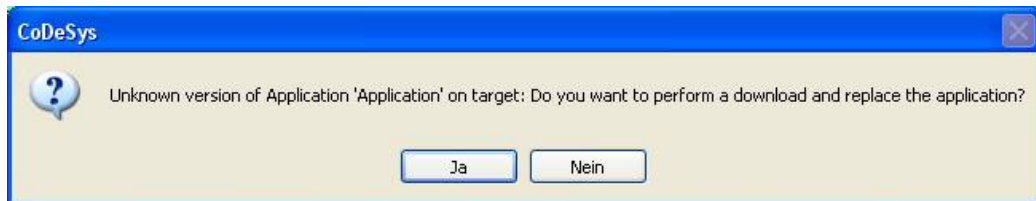
6.4 Запуск и мониторинг приложения в ПЛК

(9) Компиляция и загрузка приложения в ПЛК

Откомпилируйте данное приложение с помощью команды **Build 'Application [MyPlc: PlcLogic]'** из меню **Build** или простым нажатием <F11>. Программа будет проверена на наличие ошибок. В приведенном примере их нет, так что сообщений об ошибках быть не должно.



Теперь загрузите приложение в ПЛК: для этого используйте команду **Login** из меню **Online**. Если все настройки связи выполнены в соответствии с указаниями из пункта (8), вы получите следующее сообщение (в противном случае будет необходимо проверить настройки):



После нажатия Да (Yes) приложение будет загружено в ПЛК.

(10) Запуск приложения и мониторинг переменных

Создав проект 'Standardprojekt' и загрузив отлаженное приложение 'Application', как описано выше, вы можете запустить его на устройстве 'MyPlc'.

Запуск приложения в ПЛК:

Выполните команду **Start Application**, которая по умолчанию находится в меню **Online**. Программа начнет выполняться.

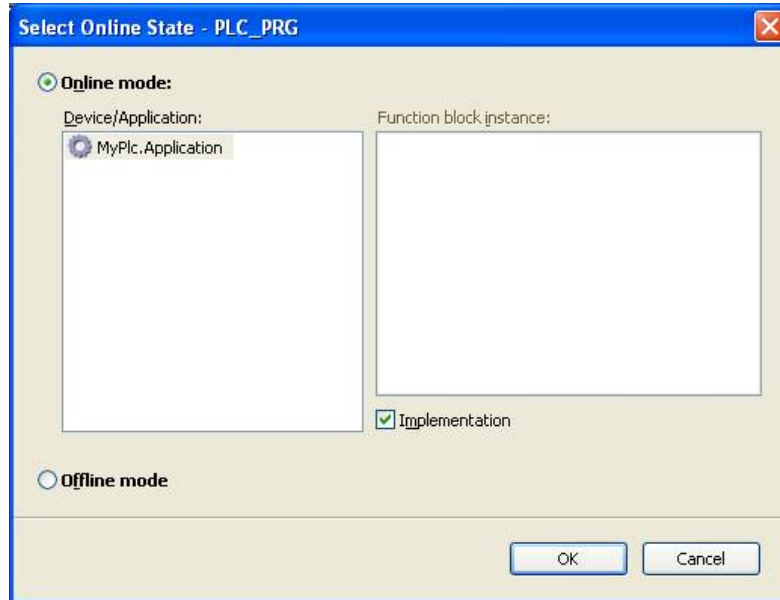
Мониторинг приложения:

Есть несколько способов мониторинга переменных прикладной программы:

1. Просмотр значений переменных определенного ROU, в окне редактора
2. Просмотр списков переменных.

1. Просмотр значений переменных определенного POU

POU может иметь несколько экземпляров, данные которых различны, поэтому при мониторинге переменных POU, нужно иметь возможность выбора необходимого экземпляра. В окне 'POUs' щелкните два раза на программе "PLC_PRG" или выберите команду 'Edit object' из контекстного меню этой строчки. В открывшемся диалоговом окне вы увидите все экземпляры PLC_PRG (в нашем примере только один):



Вы также можете выбрать, в каком режиме будет отображаться POU (онлайн или оффлайн). По умолчанию активирован онлайн режим, в котором мы и будем следить за значениями переменных. Нажмите **OK** - откроется окно PLC_PRG: нижняя часть представляет собой тело программы, мы можем непосредственно проследить за тем, как выполняется PLC_PRG; рядом с каждой переменной есть маленькое окошко с текущим значением этой переменной. Верхняя часть представляет собой таблицу, в которой перечислены элементы PLC_PRG; здесь показаны текущие значения соответствующих переменных приложения.

Expression	Type	Value
ivar	INT	140
fbinst	FB1	
in	INT	11
out	INT	13
ivar	INT	2
erg	INT	13

```

1  ivar 140 := ivar 140 +1; (* counter *)
2  fbinst(in 11 :=11); (* call function block FB1, input
3  erg 13 :=fbinst.out 13 ; (* read result from FB1 out

```

2. Запись и фиксация переменных:

Вы можете задать однократно или зафиксировать "заготовленное значение" переменной *ivar* в ПЛК; это значит, что переменной будет присваиваться данное значение в начале каждого следующего программного цикла. Щелкните два раза в поле колонки **Prepared value**, введите желаемое целое значение и закройте поле, нажав <Ввод> или щелкнув мышкой вне поля. Выполните команду '**Write values**' или '**Force values**', чтобы записать или зафиксировать это значение в ПЛК. Результат будет немедленно отображен в колонке **Value**.

3. Использование списков переменных:

Окна **Watch view** можно использовать для создания наборов объектов, мониторинг которых мы хотим выполнить при отладке.

В меню **View** выберите команду 'Watch' -> 'Watch1'. Откроется окно для ввода и редактирования списка переменных. В колонке **Expression** щелкните по первой строке таблицы, чтобы активировать поле ввода. Введите полный путь для переменной *ivar*, мониторинг которой необходимо выполнить: "*MyPlc.Application.PLC_PRG.ivar*". Нажмите <Ввод>, чтобы подтвердить ввод. Тип переменной автоматически появится в колонке **Type**.

Повторите операцию для остальных переменных. Список из примера на картинке содержит лишь объекты PLC_PRG, но вы, естественно, можете включить в набор любые переменные вашего проекта. К примеру, для FB1 вам нужно ввести "*MyPlc.Application.PLC_PRG.fbinst*".

Watch 1		
Expression	Type	Value
MyPlc.Application.PLC_PRG.ivar	INT	0
MyPlc.Application.PLC_PRG.fbinst	FB1	
MyPlc.Application.PLC_PRG.erg	INT	0

Для просмотра отдельных переменных POU используйте 'плюсик': текущее значение переменных указано в колонке '**Value**':

Watch 1		
Expression	Type	Value
MyPlc.Application.PLC_PRG.ivar	INT	264
MyPlc.Application.PLC_PRG.fbinst	FB1	
in	INT	11
out	INT	13
ivar	INT	2
MyPlc.Application.PLC_PRG.erg	INT	13

Запись и фиксация значений переменных в списках выполняются так же как было описано выше (см. п. 2).

Чтобы разорвать соединение с ПЛК, используйте команду '**Logout**' из меню **Online**.

6.5 Отладка приложения

(11) Установка шагов и точек останова

В режиме онлайн вы можете установить точки останова, в которых выполнение программы будет приостановлено.

Если программа достигла точки останова, вы можете выполнить ее по шагам. В каждой такой точке вам доступны текущие значения переменных.

Сделайте следующее:

Выделив строку 1 в PLC_PRG, нажмите <F9> или выполните команду 'New Breakpoint' из меню **Online**.

Будет установлена точка останова.

Если в данный момент приложение находится в состоянии STOP, вы увидите следующее:

```

1  ivar 2811 := ivar 2811 + 1; (* counter *)
2  fbinst(in:=11); (* call function block FB1, input parameter in=11 *)
3  erg 13 :=fbinst.out 13 ; (* read result from FB1 output "out" and write this to "erg" *)RETUR

```

Если же оно запущено, то останов будет выполнен в указанной строке:

```

1  ivar 2946 := ivar 2946 + 1; (* counter *)
2  fbinst(in:=11); (* call function block FB1, input parameter in=11 *)
3  erg 13 :=fbinst.out 13 ; (* read result from FB1 output "out" and write this to "erg" *)RETUR

```

Теперь можно выполнить программу по шагам клавишей <F8>, или используя команду **Step Into** из меню **Online**. Для выполнения программы по шагам без перехода в функциональный блок выполните команду **Step Over** (<F10>).

Вы также можете вызвать диалоговое окно Breakpoints с помощью команды 'Breakpoints' из меню **View**. Здесь можно просматривать и изменять текущие точки останова, а также установить новые.

Помните, что эти точки сохраняются в памяти, когда вы отключаетесь от ПЛК. Они будут помечены потускневшими красными кружками.

Итак, вы создали и запустили свой первый проект в CoDeSys V3. Для дальнейшего изучения возможностей системы программирования используйте встроенную систему помощи.

Желаем успехов! С радостью примем ваши замечания.

7 Удаление, настройка, восстановление

Удаление

Вы можете устанавливать более новые версии CoDeSys, не удаляя текущих. В случае необходимости деинсталляции можно использовать файл *Setup_CoDeSysV<version>.exe*. В стартовом окне выберите опцию 'Remove all installed features' и нажмите 'Next'. Кроме того можно использовать стандартную процедуру удаления программы в ОС.

Настройка

Если вы хотите изменить некоторые параметры установки продукта (устанавливаемые компоненты и т.д.), в стартовом окне мастера установки выберите опцию 'Modify'. Нажмите 'Next' и измените необходимые настройки последней установки.

Восстановление

Если по каким-либо причинам установка не была выполнена успешно, можно попробовать ее восстановить. Запустите *Setup_CoDeSysV<version>.exe* и выберите в стартовом окне опцию 'Repair'. Установка будет выполнена еще раз с прежними настройками.

8 История изменений документа

Версия	Описание	Редактор	Дата
0.1	Выпуск	MN	16.05.2006
0.2	Пример проекта	MN	26.01.2007
0.3	Доработка примера, обзор	MN/MW	08.02.2007
0.4	Исправления в обзоре	MN	09.02.2007
1.0	Релиз	MN	09.02.2007
2.0	Исправления к разделу 6 (имя устройства в нескольких местах)	MN	21.02.2007
3.0	Дополнения к разделу 6.3 (устройство по умолчанию) и 6.4 (возможности мониторинга)	MN	26.03.2007
3.0	Перевод на русский язык Проверка и коррекция текста	АГ ДП, ИП	14.05.2007