

Epsilon LD

БЫСТРЫЙ СТАРТ

DPA-310

Версия 1.1

Сентябрь 2015

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	2
УСТАНОВКА	3
Системные требования	3
Процесс установка	3
ОБЗОР СРЕДЫ РАЗРАБОТКИ.....	5
Компоненты интерфейса пользователя.....	5
Проект.....	7
POU.....	8
Устройство, дерево устройств.....	8
Приложение.....	12
Конфигурация задач	14
С ЧЕГО НАЧАТЬ.....	15
Запуск Epsilon LD	15
Создание проекта.....	16
Написание ПЛК – программы	18
Определение ресурсных объектов для запуска и контроля программы на ПЛК	20
Запуск и наблюдение за приложением на ПЛК.....	23
КОНФИГУРИРОВАНИЕ АППАРАТНОЙ ЧАСТИ	
КОНТРОЛЛЕРА REGUL R600.....	27
Конфигурирование структуры кольцевой шины	28
Конфигурирование крейтов.....	31
Конфигурирование модулей	36
КОНФИГУРИРОВАНИЕ АППАРАТНОЙ ЧАСТИ	
КОНТРОЛЛЕРА REGUL R500.....	40
Конфигурирование структуры кольцевой шины	41
Конфигурирование крейтов.....	44
Конфигурирование модулей	48
ОБЗОР ДОПОЛНИТЕЛЬНЫХ ИНСТРУМЕНТОВ.....	49
Сканер сети	49
Диагностика ПЛК.....	50
Настройка времени	53
Настройка Modbus RTU Master.....	54
ПРИЛОЖЕНИЕ А. ПАРАМЕТРЫ МОДУЛЕЙ	60
ПРИЛОЖЕНИЕ Б. ВХОДЫ/ВЫХОДЫ МОДУЛЕЙ.....	67

ВВЕДЕНИЕ

Настоящий документ содержит сведения об установке и работе в среде разработки Epsilon LD. Данное программное обеспечение предназначено для конфигурирования и программирования промышленных логических контроллеров серии «Regul».

Epsilon LD позволяет осуществлять аппаратное конфигурирование контроллеров «Regul», создание и редактирование прикладного программного обеспечения, настройку резервирования, загрузку и выгрузку проектов, пошаговую отладку прикладной программы, мониторинг работы контроллера.

Epsilon LD позволяет работать в редакторах стандарта МЭК 61131-3:

FBD - функциональные блочные диаграммы;

LD - релейно-контактная логика;

ST - структурированный текст;

IL - список инструкций;

SFC - последовательные функциональные диаграммы.

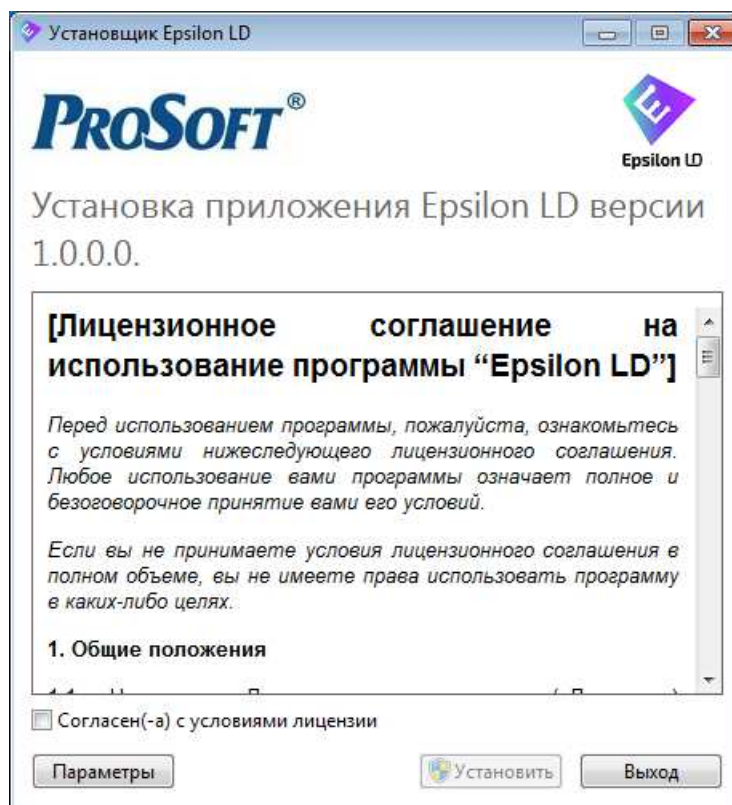
УСТАНОВКА

Системные требования

- Операционная система Windows XP или выше
- 1024 Мб оперативной памяти
- 1 Гб свободного места на HDD
- Процессор: Pentium V, Centrino > 3,0 GHz, Pentium M > 1,5GHz

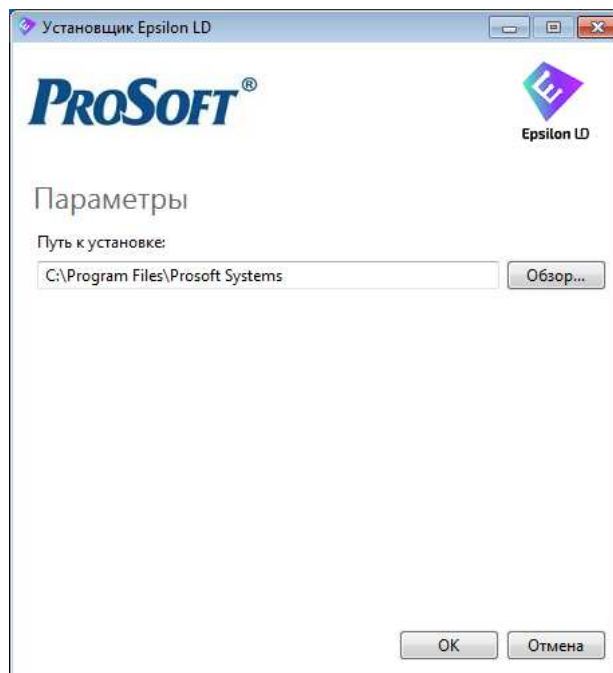
Процесс установки

1. Запустите файл Setup.exe для начала процесса установки.



Установка

2. В окне «Параметры» выберите директорию для установки.



Окно «Параметры»

3. Примите лицензионное соглашение и нажмите кнопку «Установить».
4. В процессе инсталляции может понадобиться установить дополнительное ПО (.Net Framework, VC 2005, VC 2010, VC 2013, CodeMeter), следуйте инструкциям на экране.
5. После завершения установки Epsilon LD будет запущен Package Manager (Менеджер пакетов) в консольном режиме для установки пакета расширения. Дождитесь окончания установки пакета.

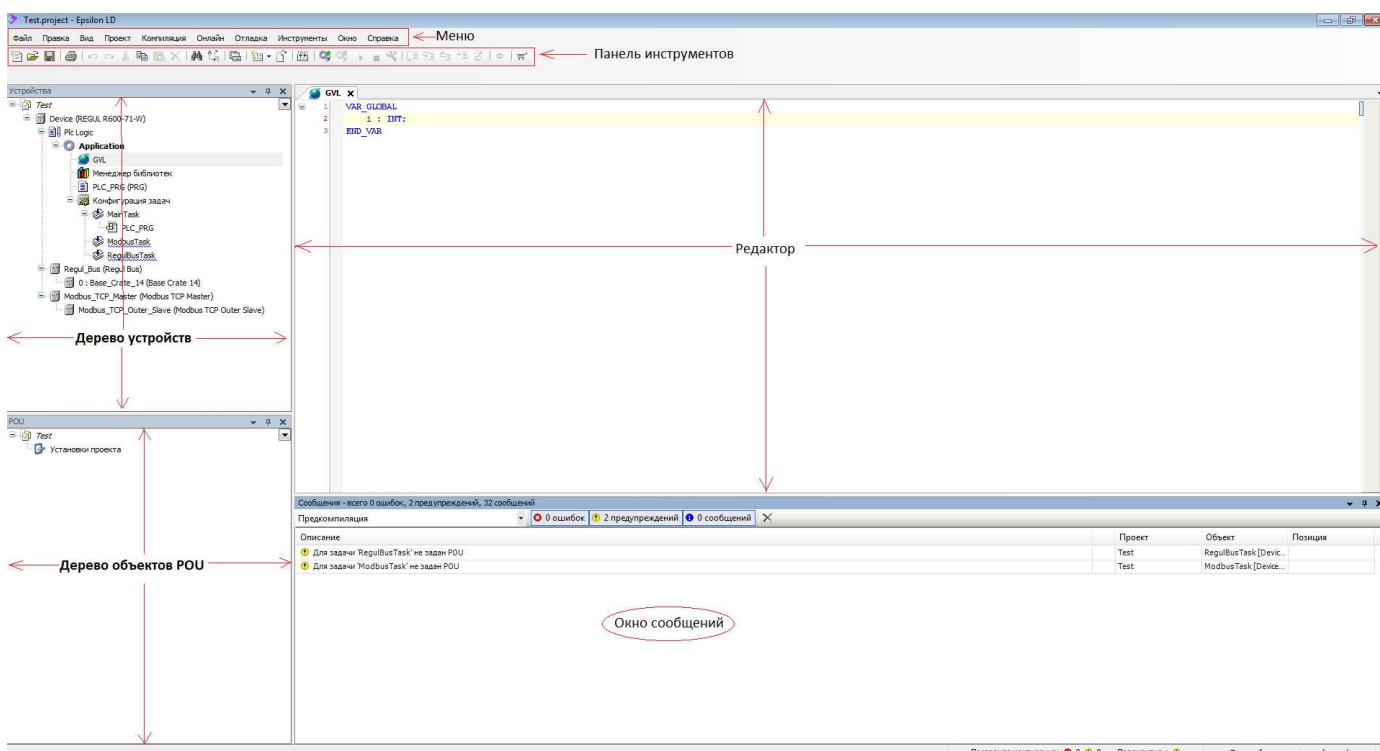
ОБЗОР СРЕДЫ РАЗРАБОТКИ

Компоненты интерфейса пользователя

Пользовательский интерфейс Epsilon LD – это расположение его «компонентов».

Внешний вид интерфейса пользователя, зависит от его настроек, которые хранятся в специальном файле и определяют набор доступных меню, панелей инструментов и возможности использования клавиатуры. Текущие настройки можно просмотреть и изменить в диалоге «Настройка», который по умолчанию открывается из меню «Инструменты».

Интерфейс пользователя включает в себя меню, панели инструментов, редактор, организатор объектов, окна наблюдений и сообщений, а также строку состояния.



Вариант пользовательского интерфейса Epsilon LD

Стандартные компоненты:

- Строка меню: содержит меню, включающие в себя все доступные команды, заданные в диалоге «Настроить». По умолчанию включена.
- Панель инструментов: содержит кнопки всех инструментов, заданных в диалоге «Настроить» или доступных за счет плагина. По умолчанию включена.

- Окно панели инструментов: содержит инструменты конкретных редакторов (открывается из меню «Вид»).
- Окно «POU»: служит для организации программных компонентов (POU, DUT и т.д.) проекта в виде дерева (открывается из меню Вид).
- Окно «Устройства»: служит для организации ресурсных объектов проекта в виде дерева (открывается из меню Вид).
- Окно редактора: используется для создания конкретного объекта в соответствующем редакторе. Окно языкового редактора (например, ST-редактор, CFC-редактор) обычно содержит языковой редактор в нижней части и редактор объявления в верхней части. Другие редакторы могут также включать диалоги (например, «Редактор задач», «Редактор устройств»). Имя POU или ресурсного объекта всегда отображается в строке заголовка окна. Открыть объект в окне редактора можно командой «Редактировать объект».

О том, что происходит с вашим проектом в офлайн (без подключения к ПЛК) или онлайн-режиме (при подключении к контроллеру), можно узнать с помощью следующих компонентов:

- Окно сообщений: в этом окне отображаются сообщения компиляции, предкомпиляции, загрузки и т.д.
- Окна наблюдения и онлайн просмотр редакторов: обеспечивают просмотр POU или списков наблюдаемых выражений, заданных пользователем.
- Строка состояния: строка на нижней границе пользовательского интерфейса содержит информацию о текущем пользователе. Если в данный момент вы работаете в окне редактора, то отображается также текущая позиция курсора и режим редактирования. В режиме онлайн отображается текущее состояние программы.

Информация онлайн-режима:

- RUN – программа запущена.
- STOP – программа остановлена.
- HALT ON BP – программа остановлена в точке останова.
- Program loaded – программа загружена на контроллер.

- Program unchanged – программа в контроллере идентична программе в системе программирования.
- Program modified (Online Change) – программа в контроллере отличается от программы в системе программирования, необходимо онлайн-изменение.
- Program modified (Full download) – программа в контроллере отличается от программы в системе программирования, требуется полная загрузка.

Окна, которые отображаются внутри или рядом с обрамляющим окном пользовательского интерфейса бывают двух типов:

1. Одни из них можно прикрепить к любой из границ обрамляющего окна или передвигать по экрану независимо от него. Кроме того, их можно «спрятать». В этом случае они будут представлены в виде вкладок на границе окна. Такие окна содержат информацию, которая не зависит от отдельных объектов проекта (окно сообщений, ROU, устройства, окно инструментария). Доступ к ним может осуществляться через меню 'Вид'. Большинство таких окон включают неизменяемую панель инструментов с кнопками для сортировки, просмотра и поиска.
2. Другие открываются при просмотре или редактировании конкретных объектов проекта в соответствующем редакторе. Они отображаются в области редактора, снабженного закладками, или, в зависимости от настроек интерфейса, в виде MDI-окон. Их нельзя «спрятать» или «отсоединить» от обрамляющего окна. Доступ к ним может осуществляться через меню «Окно».

Проект

Проект содержит объекты ROU (функции, глобальные переменные, функциональные блоки и т.д., подробнее см. далее), составляющие ПЛК-программу, а также определения ресурсов (устройства, приложения, конфигурация задач), необходимые для запуска одного или нескольких экземпляров программы (приложения) в конкретной целевой системе (ПЛК, устройстве). Для работы с объектами ROU используется окно просмотра ROU, для работы с ресурсами, зависящими от устройства, используется окно просмотра «Устройства» (подробнее в главе «Устройство, дерево устройств»). Проект сохраняется в файле <имя проекта>.project.

Конфигурация проекта может быть выполнена в диалогах «Настройка» и «Опции» в меню «Инструменты». Базовая конфигурация нового проекта (структура меню, заданные

объекты) определяется текущим шаблоном проекта. Шаблон выбирается при создании нового проекта командой «Создать проект».

Для задания установок проекта используется объект «Установки проекта», который по умолчанию отображается в окне «POU». Также существует объект «Информация о проекте», который добавляется в окно «POU» сразу после того, как он был вызван соответствующей командой меню (обычно меню Проект). Диалог «Информация о проекте» можно использовать для редактирования или просмотра специфической информации о проекте, такой как данные файла, статистику объектов, имя автора.

POU

POU (Program Organization Unit) – компонент организации программы. В большинстве случаев под этим термином понимается любой объект, являющийся составной частью ПЛК-программы.

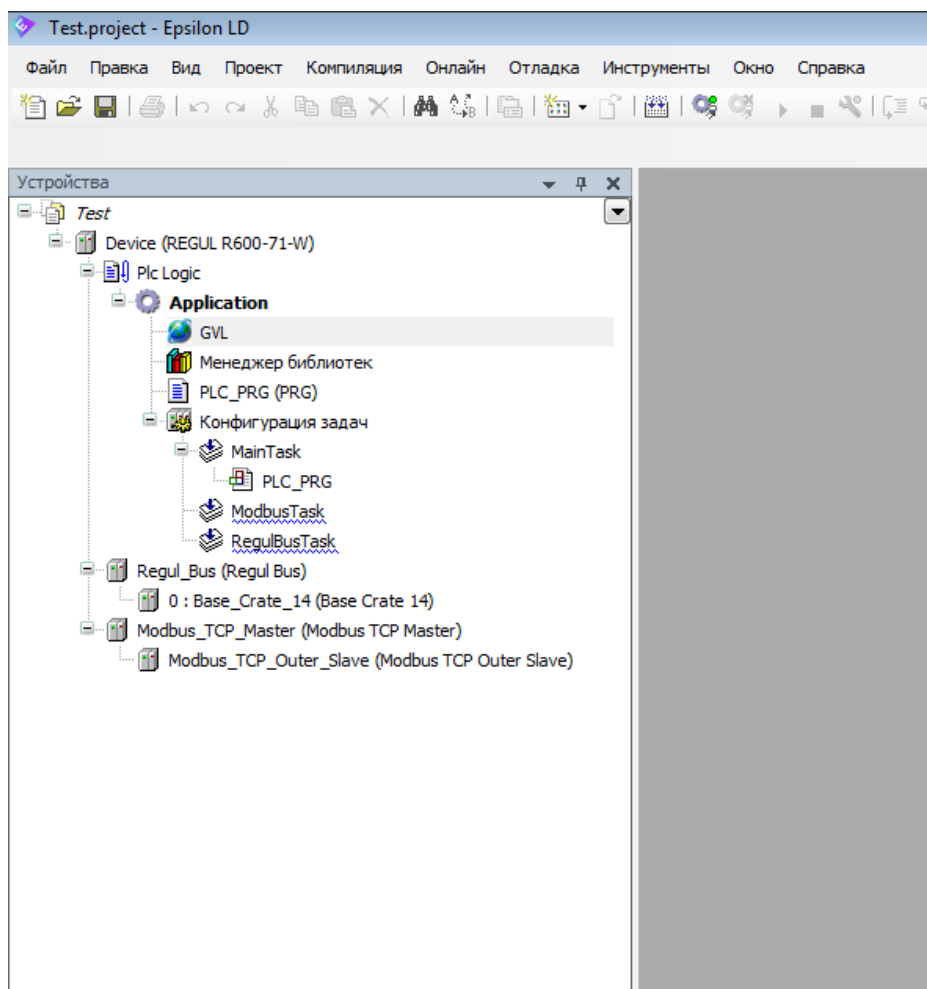
POU, которые представлены в окне "POU", не зависят от устройства. Однако их экземпляры могут быть дополнительно ориентированы на применение в конкретном приложении (устройстве). В этом случае POU должен вызываться задачей соответствующего приложения.

POU, которые отображаются только в окне «Устройства», т.е. располагаются в дереве устройств непосредственно под приложением, могут быть использованы только приложениями, которые находятся под данным приложением («дочерними»).

Просмотр и изменение выполняется в окне редактора. Объектом POU может быть программа, функция, функциональный блок, а также метод, действие, интерфейс, DUT (объект типа данных) или какой-либо внешний файл произвольного формата.

Устройство, дерево устройств

В окне Устройства («дерево устройств») вы можете определить оборудование, на котором будет запускаться приложение.



Дерево устройств

Каждое устройство представляет собой специфический (целевой) аппаратный объект. Примеры: контроллер, шинный соединитель, привод, модуль ввода/вывода, монитор. Устройство определяется его файлом описания (специальный файл в формате xml). Для того чтобы его можно было вставить в дерево устройств, оно должно быть установлено на локальной системе. В файле описания устройства задаются его свойства, касающиеся конфигурируемости, программируемости и возможности подключения к другим устройствам.

Окно Устройства («дерево устройств») содержит не только устройства, но и все объекты, необходимые для запуска приложения на устройстве (контроллере, ПЛК). Их также можно назвать «ресурсами». К ним относятся такие объекты, как «устройство», «приложение», «конфигурация задач» и «задача». Даже такие, чисто программные объекты, как ROU, списки глобальных переменных и «Менеджер библиотек», могут отображаться не в окне ROU в качестве шаблонов для экземпляров всего проекта, а только в окне устройств. В этом случае они доступны только для «своего» приложения или для «дочерних приложений» данного приложения.

Общие понятия:

- Корневым узлом дерева всегда является символический элемент: 
- Конфигурация ПЛК определяется топологической структурой устройств в дереве устройств. В отдельных диалогах выполняется лишь конфигурация конкретных устройств или параметров задачи. Таким образом, аппаратная структура представлена в дереве устройств соответствующим расположением объектов-устройств.
- Каждый объект дерева устройств представлен символом, символьным именем (редактируется) и типом устройства (т.е. имя устройства, заданное в описании устройства).
- Устройство может быть программируемым или просто настраиваемым. Тип устройства определяет его возможную позицию в дереве ресурсов, а также какие ресурсы можно вставлять под данным устройством. Программируемые устройства обозначаются дополнительным узлом «Plc Logic», который автоматически вставляется под устройством. Под этим узлом можно вставлять объекты, необходимые для программирования устройства (приложения, списки текстов и т.д.), а также функциональные объекты, такие как «Менеджер параметров».
- В одном проекте может использоваться несколько программируемых устройств независимо от типа.
- Конфигурация устройства, касающаяся соединения, параметров, соотнесения входов/выходов, выполняется в диалоге устройства (редакторе устройства), который открывается при двойном щелчке по устройству в окне устройств

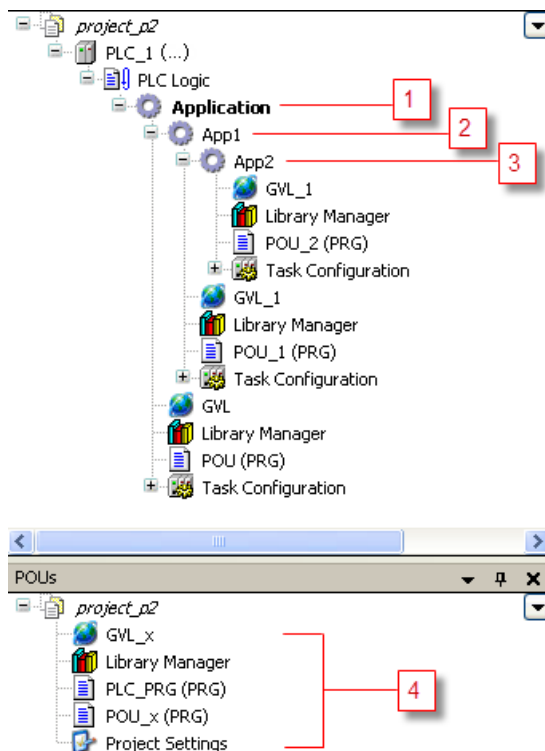
В режиме онлайн (при подключении к ПЛК) иконка рядом с устройством указывает на его состояние:

- : устройство запущено;
- : устройство пока не запущено или не настроено, либо содержит ошибки;
- : устройство запущено, доступна диагностическая информация;
- : устройство настроено, но не запущено;
- : ПЛК остановлен, I/O(входа/выхода) - обновление в остановленном режиме отключено.

Организация и конфигурирование объектов в дереве устройств, правила:

- Для добавления объекта используется команда «Добавить объект». Чтобы добавить непосредственно объект «устройство» можно использовать команду «Добавить устройство». В любом случае возможные типы объекта зависят от текущей выбранной позиции в дереве.
- Изменить позицию объекта можно с помощью стандартных команд категории «Буфер обмена» (Вырезать, Копировать, Вставить, удалить) или перетаскив выбранный объект с зажатой кнопкой мыши и клавишей <Ctrl>. В случае, если вставляемый объект может быть помещен как ниже текущего элемента, так и над ним, откроется диалог «Выберите позицию вставки», в котором необходимо указать позицию для вставки объекта. При добавлении устройств с помощью функции «Копировать и вставить» новое устройство получает то же имя, к которому присоединяется номер по возрастанию
- Устройство, вставленное в дерево устройств, может быть заменено другой версией устройства того же типа или устройством другого типа (обновление устройства). Дерево конфигурации, расположенное под соответствующим устройством, по мере возможности, не изменится.
- Только объекты «устройство» могут помещаться непосредственно под корневым узлом с именем проекта. Если в диалоге «Добавить объект» вы выбрали другой тип объекта, например, «Список текстов», он будет добавлен в окно ROU! Если не выбрано ни одного элемента будет автоматически выбран корневой узел.
- Объект «приложение» может быть вставлен только под узлом «Plc Logic» (символьный узел для программируемых устройств). При вставке нескольких приложений или дерева приложений, имена всех приложений устройства должны быть уникальными. Под приложениями могут быть вставлены другие программные объекты, такие как DUT, GVL (список глобальных переменных), или объекты визуализации. Конфигурация задач должна быть вставлена под каждым приложением. В ней задаются соответствующие вызовы программ (экземпляры ROU из окна ROU или специфические ROU приложения). Если непосредственно под устройством вставлено несколько приложений, в отображении входов/выходов устройства необходимо указать, какое именно приложение имеется в виду.
- Приложение может быть вставлено под другим приложением: итоговую конструкцию будем называть «Материнское приложение» – «Дочернее приложение». Дочернее

приложение может использовать объекты материнского приложения, но не наоборот! Требование: дочернее приложение всегда может быть удалено или заменено, не затрагивая материнское приложение. См. пример ниже: «Application» является материнским приложением для «App1», который в свою очередь является материнским приложением для «App2»:



«Материнские» и «дочерние» приложения

1. Доступ к «App1», «App2», их POU, глобальным переменным и библиотекам невозможен.
2. Возможен доступ к «Application» и его GVL (список глобальных переменных), POU и библиотекам.
3. Возможен доступ к «Application» и «App1», а также к их GVL, POU и библиотекам, для них приложениями могут создаваться экземпляры.
4. Для них приложениями могут создаваться экземпляры.

Приложение

«Приложение» (Application) – это набор объектов, необходимых для запуска конкретного экземпляра ПЛК-программы на конкретном устройстве (ПЛК, контроллер). В связи с этим для «независимых» объектов из окна POU создаются экземпляры, которые

назначаются устройствам в окне устройств, что отвечает требованиям объектно-ориентированного программирования.

Основные моменты работы с приложениями:

- Приложение представлено объектом «Приложение» в дереве устройств. Он может быть вставлен под узлом «Plc Logic» (программируемое устройство). Под ним вставляются объекты, определяющие «набор ресурсов».
- Все приложения под одним устройством должны иметь уникальные имена.
- Прежде чем вставить приложение с помощью команды «Добавить объект», выберите соответствующее программируемое устройство в окне устройств. Иначе в диалоге «Добавить объект» вы не сможете выбрать тип объекта «приложение». В диалоге введите имя приложения и нажмите кнопку «Добавить». Оно будет добавлено в дерево устройств под выбранным устройством.
- Неотъемлемой частью каждого приложения является «Конфигурация задач», управляющая выполнением программы (экземпляры ROU, ROU для конкретных приложений). Кроме того, ей можно назначить ресурсные объекты, такие как списки глобальных переменных, библиотеки и т.д. В отличие от тех, что расположены в окне ROU, они могут использоваться только конкретным приложением и его «дочерними» компонентами.
- В проекте может быть задано несколько приложений для одного устройства. «Устройство» представляет собой специфическое аппаратное средство, на котором должна запускаться ПЛК-программа (приложение). В окне устройств оно представлено объектом – «ресурсом», т.е. они назначены и будут выполняться на одном контроллере. Таким образом, возможна вложенная организация приложений («родительские» и «дочерние приложения»). В этом случае объекты, назначенные «родительскому приложению», могут использоваться «дочерними компонентами», но не наоборот.
- Если непосредственно под устройством вставлено несколько приложений, в соотнесении входов/выходов устройства необходимо указать, какое именно приложение имеется в виду.
- Приложение, с которым вы собираетесь работать, должно иметь состояние «активное приложение».

- При попытке загрузить приложение на целевое устройство (ПЛК или эмулирующее устройство) будет выполнена проверка текущих приложений на ПЛК, а также проверка на соответствие параметров приложения на ПЛК и в конфигурации проекта. При несоответствии будут выдаваться сообщения, после чего можно будет удалить приложения с ПЛК.

Конфигурация задач

В «Конфигурации задач» определяется одна или несколько задач, контролирующих выполнение прикладной программы. Она является неотъемлемым ресурсным объектом для приложения и должна быть помещена под приложением в дереве устройств. Задача может вызывать как ROU для конкретного приложения, так и программы, расположенные в окне ROU. Во втором случае для программы, доступной для всего проекта, в приложении будет создан его экземпляр. Конфигурацию задач можно изменять в Редакторе задач, доступные опции зависят от устройства.

В режиме онлайн «Редактор задач» предоставляет окно мониторинга, содержащее информацию о циклах, временах циклов и состоянии задач.

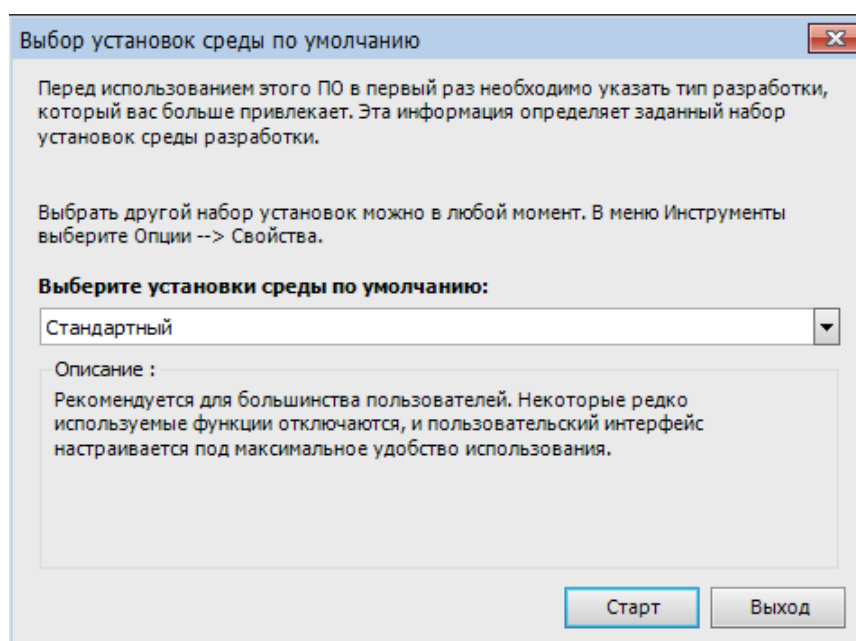
Еще одной возможностью «Конфигурации задач» (если поддерживается устройством) является функция «профилирования», выполняющая динамический анализ ROU, контролируемых конкретной задачей. Она предоставляет информацию о времени выполнения, количестве вызовов ROU и строках кода, необрабатываемых на данный момент.

С ЧЕГО НАЧАТЬ

Запуск Epsilon LD

Для запуска программы, выберите в меню «Пуск» следующее: Программы > Prosoft-Systems > Epsilon LD. Также можно воспользоваться ярлыком на рабочем столе, который будет автоматически создан после установки программы.

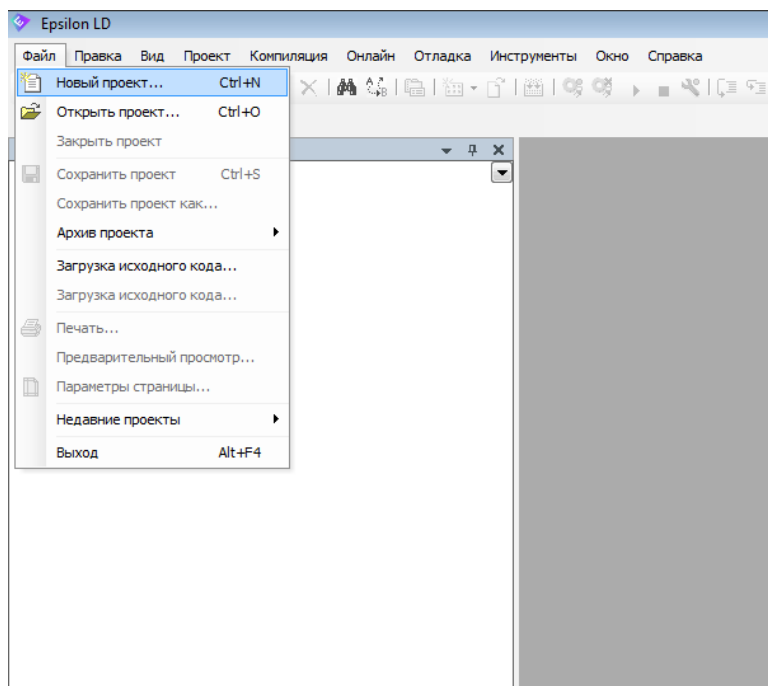
При первом запуске программы вам будет предложено выбрать набор настроек по умолчанию. Выберите «Стандартный» тип настроек. Вы всегда сможете изменить установки среды на «Профессиональные» из меню «Инструменты» > «Опции» > «Свойства» > «Заданные наборы свойств». При выборе «Стандартного» набора настроек среда разработки адаптируется для удобства пользователя, некоторые редко используемые функции будут отключены.



Выбор установок среды по умолчанию

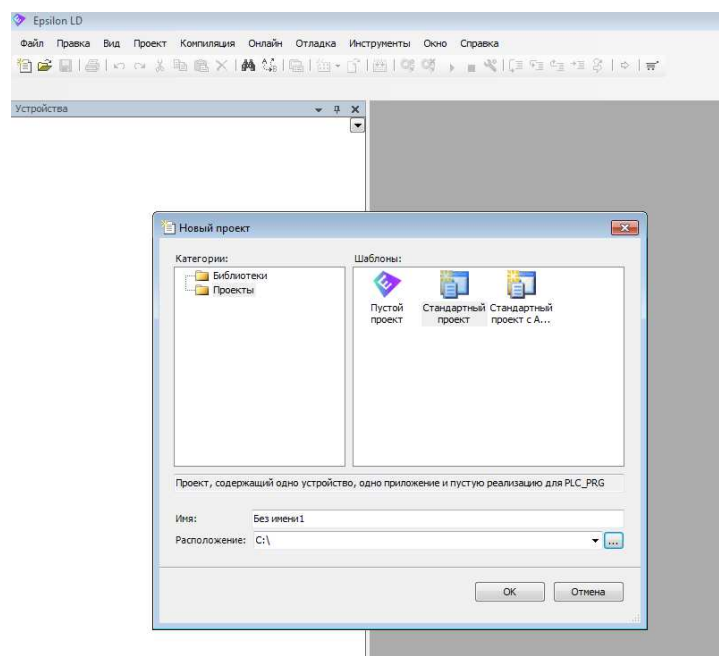
Создание проекта

Для создания нового проекта в меню «Файл» выберите команду «Новый проект»:



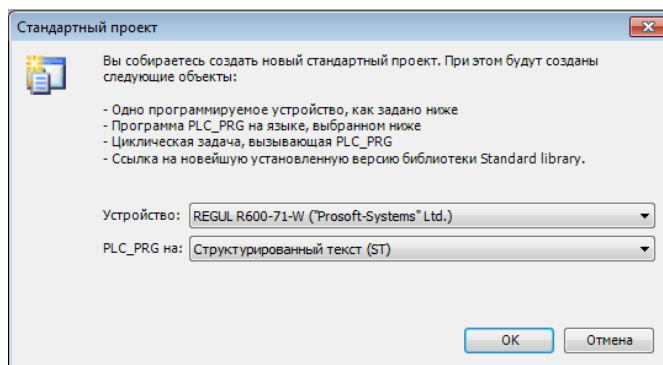
Команда «Новый проект»

Откроется диалоговое окно создания нового проекта.



Диалоговое окно «Новый проект»

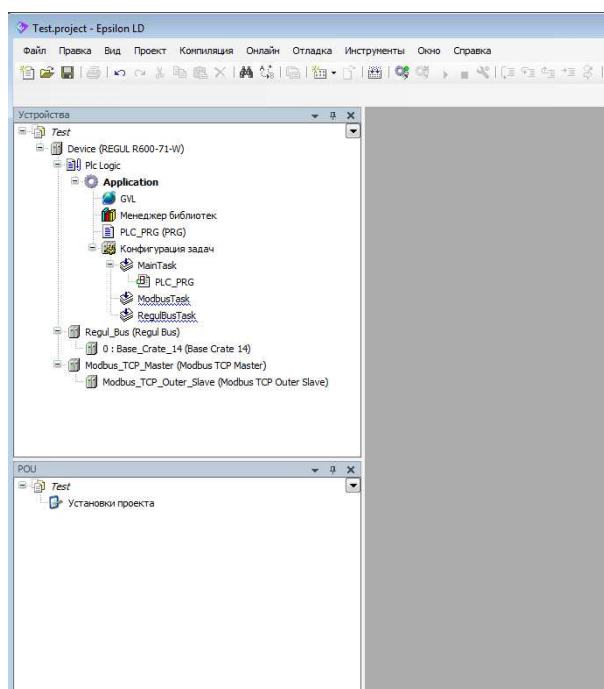
В данном окне в поле «Шаблоны» выберите «Стандартный проект» и введите имя нового проекта и путь к директории файла проекта. Для подтверждения нажмите кнопку «ОК». Откроется диалог мастера создания нового проекта.



Мастер создания нового проекта

Выберите нужное вам устройство и язык программирования, затем нажмите кнопку «ОК». Теперь имя проекта появится в строке заголовка интерфейса и в качестве символического корневого узла в окнах POU и «Устройства».

В окне «Устройства» отображается дерево с устройством «Device» и присоединенным к нему ниже приложением. Это приложение содержит программу «PLC_PRG» на языке ST и обязательную Конфигурацию задач, включающую в себя «MainTask» для управления PLC_PRG. Символьное имя устройства можно изменить, выделив его и затем нажав кнопку «пробел». Откроется рамка ввода, в которой можно ввести новое имя.

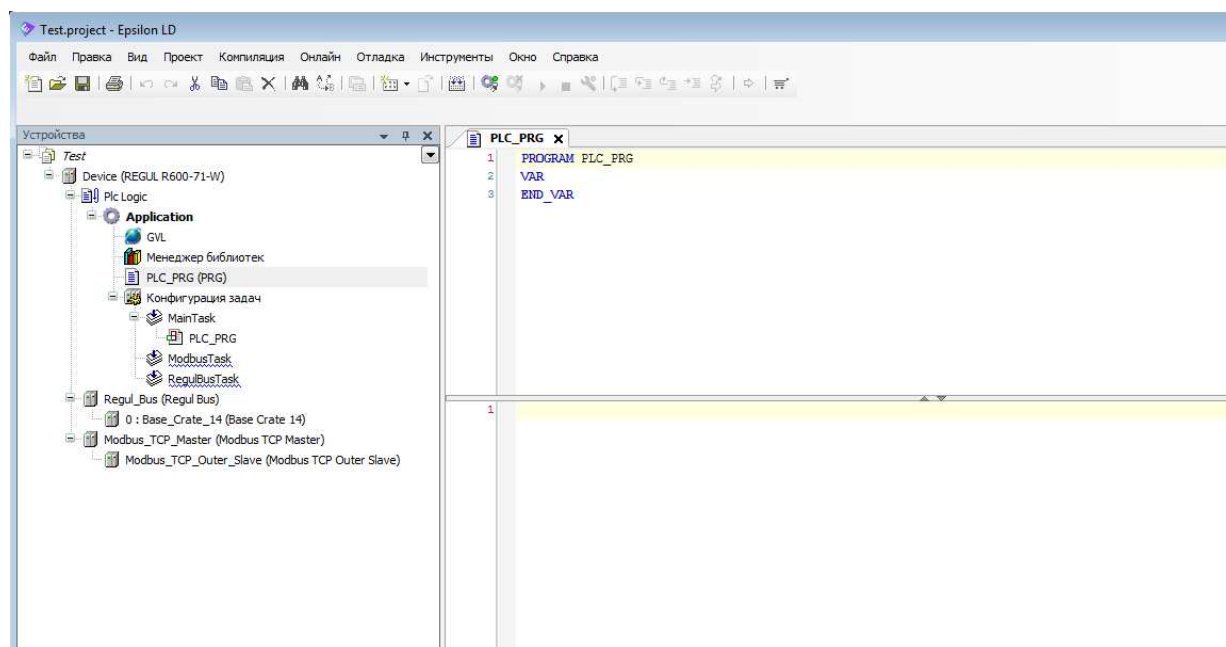


Окна «POU» и «Устройства»

Написание ПЛК – программы

Объявление переменных в PLC_PRG

В окне редактора ST откройте POU "PLC_PRG", который по умолчанию доступен в окне «Устройства». Для этого щелкните два раза мышкой на объекте "PLC_PRG" в дереве устройств. POU "PLC_PRG", который по умолчанию доступен в окне «Устройства», автоматически откроется в окне редактора ST в центре интерфейса Epsilon LD. По существу, любой POU, доступный в дереве POU или устройств, можно открыть двойным щелчком мыши. ST-редактор состоит из раздела объявления (выше) и "тела" (раздела реализации). В разделе объявления отображаются номера строк, тип и имя POU ("PROGRAM PLC_PRG"), а также ключевые слова "VAR" и "END_VAR" для объявления переменных. Тело пусто, содержит пока только номер первой строки.



Редактор программы PLC_PRG

В разделе объявления поместите курсор после VAR и нажмите клавишу <Ввод>. Будет вставлена пустая строка, в которой вы можете объявить переменные «ivar» и «erg» типа INT и «fbinst» типа FB1:

```
PROGRAM PLC_PRG
```

```
VAR
```

```
    ivar: INT;
```

```
    fbinst: FB1;
```

```
    erg: INT;
```

```
END_VAR
```

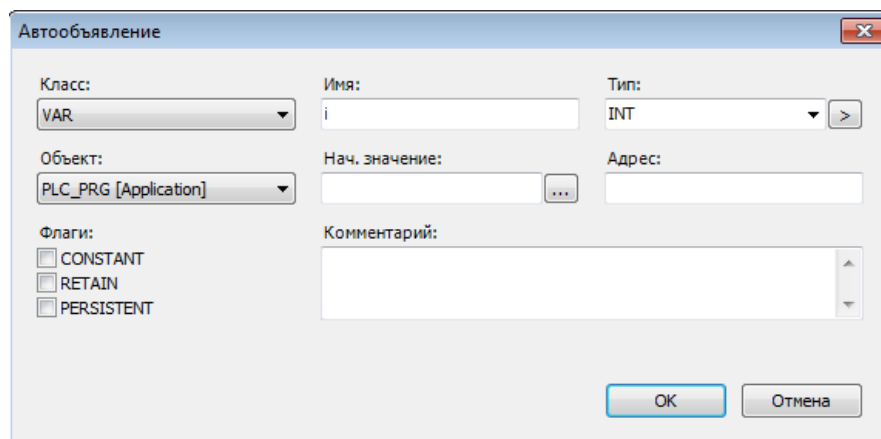
Помимо этого, можно ввести инструкцию непосредственно в разделе реализации и применить функцию «Автообъявления».

Ввод программного кода в тело PLC_PRG

Введите следующий код в тело программы (после разделительной черты):

```
ivar:= ivar+1;           // счетчик
fbinst(in:=11, out=>erg); // вызов функционального блока типа FB1, с входным параметром "in",
                        // выход записывается в "erg".
```

Также можно воспользоваться функцией Автообъявления: без какого-либо предварительного ввода в разделе объявления введите инструкцию непосредственно в тело программы и нажмите клавишу «Ввод». Для каждой необъявленной переменной будет открываться диалог Автообъявления, где вы можете задать параметры объявления:



Автообъявление

Создание дополнительного программного ROU

Для создания нового функционального блока выполните следующие шаги:

1. Выделите левой кнопкой мыши узел «Application» в дереве устройств
2. Выберите команду «Добавить объект» из меню Проект.
3. В левой части диалога «Добавить объект» и выберите опцию Функциональный блок в разделе «Тип».
4. Введите имя функционального блока «FB1»
5. В качестве языка реализации выберите «Структурированный текст (ST)».
6. Для подтверждения установок объекта нажмите кнопку «Добавить».

После этого откроется другое окно редактора для нового функционального блока. Здесь, по аналогии с PLC_PRG, объявите следующие переменные:

```
FUNCTION_BLOCK FB1
```

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

В разделе реализации введите следующее:

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

Определение ресурсных объектов для запуска и контроля программы на ПЛК

Запуск Gateway - сервера и виртуального ПЛК

Gateway - сервер автоматически запускается в качестве сервиса при запуске системы. Убедитесь, что на панели задач есть иконка , указывающая на то, что сервер запущен. Иконка  говорит о том, что gateway на данный момент остановлен. Эта иконка является частью программы GatewaySysTray, предназначенной для контроля и наблюдения за сервисом Gateway. Она включает в себя меню, содержащее команды start и stop, что позволяет пользователю останавливать и перезапускать сервис вручную. Меню также содержит команду Exit Gateway Control, которая закрывает только программу GatewaySysTray, но не сервис Gateway. Программа GatewaySysTray запускается автоматически при запуске Windows. Однако ее можно также запустить из меню Программы.

При отсутствии реального ПЛК, протестировать прикладную программу можно при помощи виртуального контроллера Control Win V3. Данный ПЛК доступен в качестве сервиса после запуска системы. На панели задач он представлен иконкой:  для состояния «остановлен»,  для состояния «запущен». ПЛК-сервис может автоматически запускаться при запуске системы, если это поддерживается самой системой. В противном случае для его запуска необходимо вручную применить команду «Start PLC» из меню, которое открывается щелчком мыши по иконке. Для использования виртуального контроллера в проект следует добавить устройство Control Win V3.

Установка «Активного приложения»

В окне устройств имя «Application» выделено жирным шрифтом. Это означает, что данное приложение имеет статус "активного приложения". Таким образом, все команды и

действия, связанные с соединением с ПЛК, будут относиться к этому приложению. Для того чтобы задать приложение как "активное", кликните правой кнопкой мыши по требуемому приложению и выберите команду «Установить активное приложение» из контекстного меню.

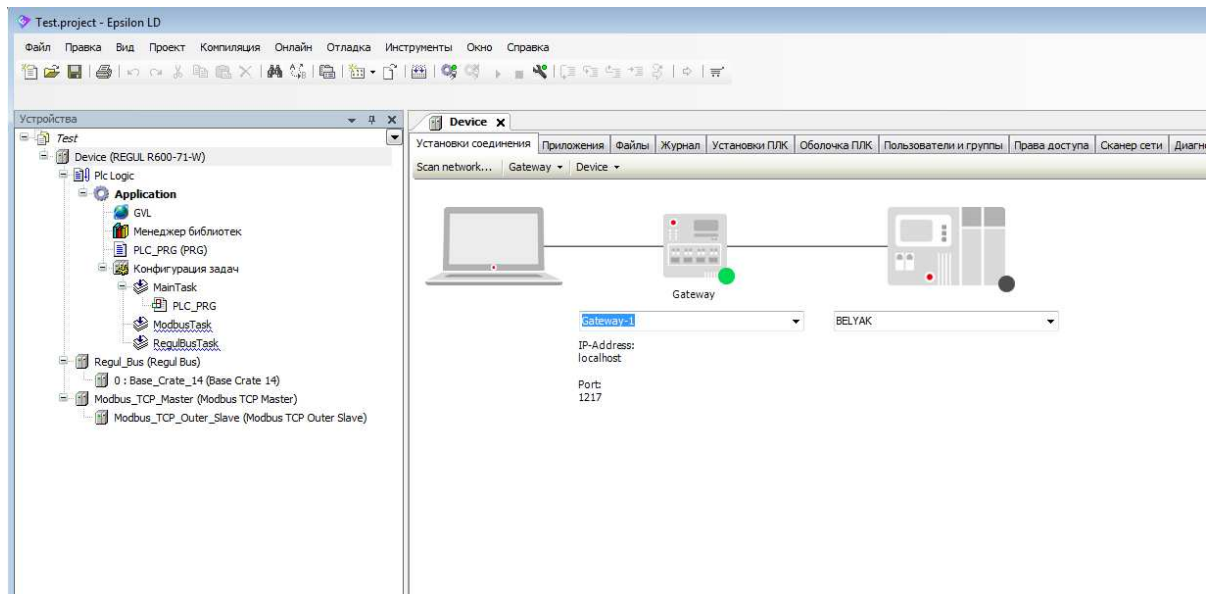
Конфигурация канала связи с ПЛК

Двойным щелчком по устройству в окне устройств можно открыть стандартный редактор устройства со вкладкой «Установки соединения». Здесь вам необходимо установить соединение между ПЛК (целевым устройством) и системой программирования. Сначала нужно выбрать gateway из списка или создать новый. Обратите внимание, что локальный gateway-шлюз будет уже доступен, он автоматически запускается при старте сервиса Gateway, с помощью программы GatewaySysTray (см. главу «Запуск Gateway-сервера и ПЛК»).

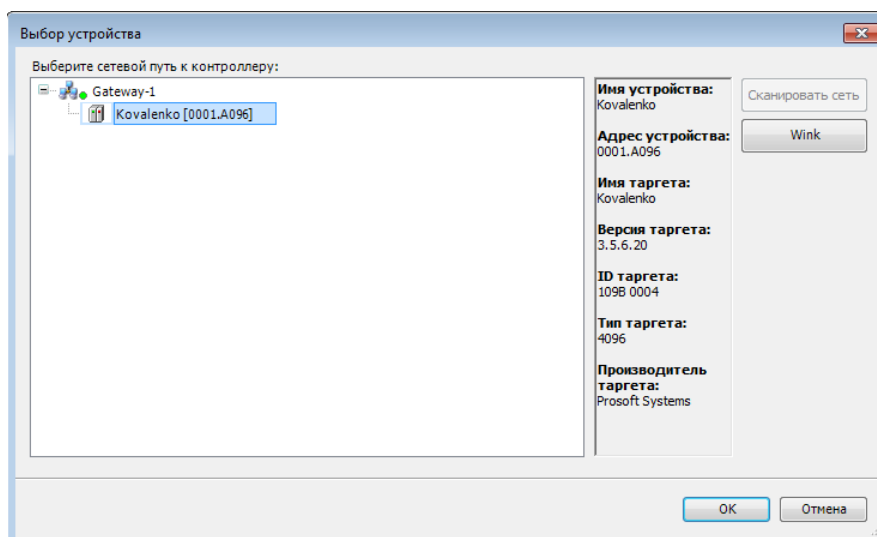
Для добавления нового gateway выполните команду Gateway > Add new gateway. В открывшемся окне введите символическое имя gateway, задайте тип драйвера "TCP/IP" и введите IP-адрес (для открытия рамки редактирования два раза щелкните по полю колонки). Установку «Port» оставьте без изменений (следует изменять, если стандартный порт заблокирован).

Если gateway работает нормально, то маркер рядом со схематическим изображением gateway – зеленого цвета, в противном случае – красного. Чёрный маркер означает, что соединение еще не определено.

Теперь определите канал к целевому устройству, который следует подключить через заданный gateway. Нажмите кнопку «Scan network» для поиска всех доступных устройств в вашей локальной сети. Откроется окно «Выбор устройства», все найденный ПЛК в вашей сети будут отображаться под gateway с отступом вправо. Выберите нужный ПЛК и нажмите кнопку «ОК».



Установки соединения



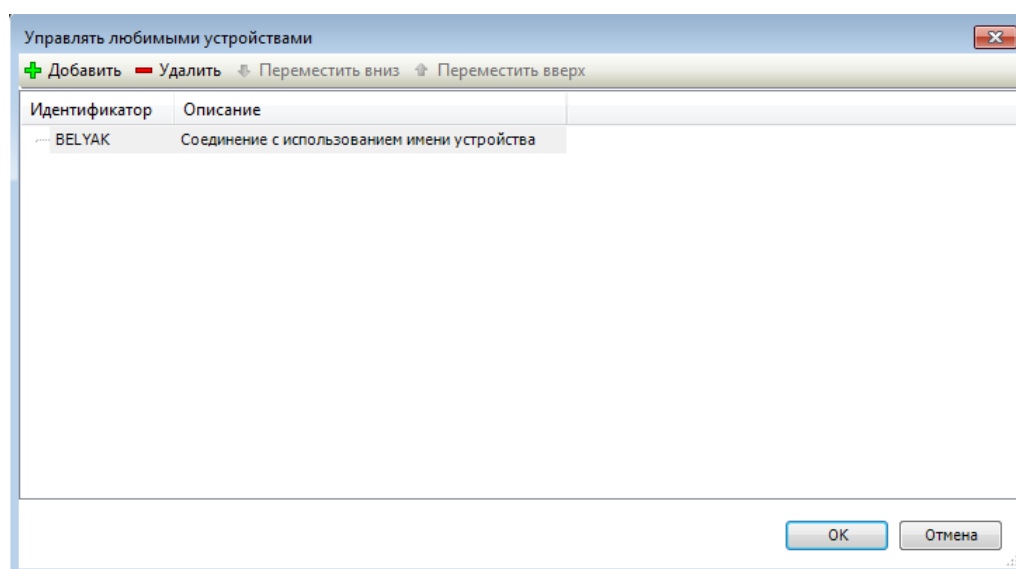
Выбор устройства

Рассмотрим параметры установки соединения расположенные в меню «Device»:

- Store communication settings in project (Сохранять настройки соединения в проекте): если опция активирована, то текущий сетевой путь будет сохраняться в проекте, а не в локальных настройках на вашем компьютере;
- Confirmed online mode (Безопасный онлайн-режим): если опция активирована, то для выполнения следующих онлайн-команд будет запрашиваться подтверждение пользователя: «Фиксировать значения», «Множественная загрузка», «Освободить фиксацию для <приложение>», «Один цикл», «Запустить <приложение>», «Остановить <приложение>», «Записать значения»;

- Filter network scans by target ID (Фильтровать сканирование сети по идентификатору устройства): если данная опция активирована, то, при сканировании сети, будут показываться только найденные устройства, идентификаторы которых совпадают с идентификатором текущего редактируемого устройства.

Команда «Управлять любимыми устройствами» меню «Device» позволяет настраивать список «любимых устройств» (список устройств, настройки подключения к которым будут сохранены, и подключаться к ним можно без сканирования сети). В данном диалоговом окне можно добавить устройство вручную, без использования механизма сканирования сети (для этого необходимо знать «имя устройства» либо адрес устройства, либо его IP адрес).



Диалоговое окно «Управлять любимыми устройствами»

Запуск и наблюдение за приложением на ПЛК

Компиляция и загрузка приложения в ПЛК

Если вы хотите просто проверить вашу «активную» прикладную программу на синтаксические ошибки, выполните команду «Компилировать приложение» (контекстное меню, меню «Компиляция») или используйте клавишу «F11». Замечание: никакого кода в этом случае не генерируется. Информация, предупреждения и сообщения об ошибках будут отображаться в окне сообщений, которое по умолчанию располагается в нижней части интерфейса.

Для загрузки приложения в контроллер выполните команду Онлайн > Логин. Для того чтобы отключиться от ПЛК, используйте команду Отключение из меню Онлайн.

Запуск и мониторинг приложения

После того, как вы создали проект и загрузили приложение в ПЛК, его можно запустить, выполнив команду «Отладка» > «Старт». В самом низу в строке состояния должна отображаться надпись «Запуск» на зеленом фоне.

Есть три варианта наблюдения за переменными прикладной программы, а также доступа к ним:

1. Онлайн-наблюдение за конкретным POU
2. Запись и фиксация переменных
3. Окна просмотра с заданными списками просмотра

1. Онлайн-наблюдение за конкретным POU

В окне экземпляра POU представлены все выражения этого экземпляра в табличном виде в разделе объявления и, если активировано, «построчный мониторинг» также в разделе реализации. Открыть окно онлайн-наблюдения можно двойным щелчком по "PLC_PRG" в окне устройств или применив команду «Редактировать объект» из контекстного меню. В нижней части окна отображаются строки кода, которые вы ввели до подключения. Рядом с каждой переменной расположено маленькое окошко мониторинга, в котором отображается текущее значение переменной (встроенный мониторинг). В верхней части таблицы представлены наблюдаемые выражения POU, т.е. текущие значения соответствующих переменных приложения "Application" на ПЛК.

Expression	Type	Value	Prepared value
ivar	INT	43	
fbinst	FB1		
erg	INT	13	

```

1  ivar 43 := ivar 43 + 1; // counter
2  fbinst(in 11 := 11, out 13 => erg 13); // call function block of type FB1
3
4  // with input parameter "in"
5  // output is written to "erg" RETURN

```

Наблюдение за POU

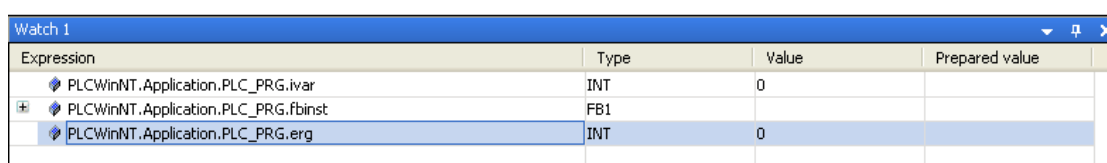
2. Запись и фиксация переменных

Можно записать или фиксировать "Подготовленное значение" переменной *ivar* на ПЛК. Это значит, что данное значение будет присвоено переменной *ivar* в начале следующего цикла. После двойного щелчка по полю в колонке «Подготовленное значение» введите нужное целочисленное значение и нажмите «Ввод» или щелкните мышкой за пределами

поля. Выполните команду «Записать значения» или «Фиксировать значения» для записи или фиксации этого значения на ПЛК. В колонке «Значение» вы сразу увидите результат.

3. Окна просмотра

Окна просмотра можно использовать для задания особых наборов наблюдаемых выражений приложения, например, с целью отладки. Используйте команду Просмотр > Watch1 из меню «Вид». Откроется окно просмотра. В колонке «Выражение» щелкните мышкой по первой строке таблицы, чтобы открыть рамку редактирования. Введите полный путь к переменной, за которой следует установить мониторинг, например «PLCWinNT.Application.PLC_PRG.ivar». Рекомендуется использовать «Ассистент ввода». Закройте рамку клавишей «Ввод». Тип будет добавлен автоматически.



Expression	Type	Value	Prepared value
PLCWinNT.Application.PLC_PRG.ivar	INT	0	
PLCWinNT.Application.PLC_PRG.fbinst	FB1		
PLCWinNT.Application.PLC_PRG.erg	INT	0	

Пример окна просмотра

Теперь выполните команду Старт из меню Онлайн (если вы этого еще не сделали). Приложение запустится на ПЛК, а текущее значение будет отображаться в колонке «Значение».

Запись и фиксация переменных здесь также доступны, как описано в пункте 2. Для того чтобы отключиться от ПЛК, используйте команду «Отключение» из меню «Онлайн».

Задание точек останова и пошаговое выполнение программы

Для приостановки выполнения программы в конкретном месте в режиме онлайн можно задать точку останова. После того, как программа достигла точки останова, ее выполнение можно продолжить по шагам. В каждой точке можно просмотреть значения переменных.

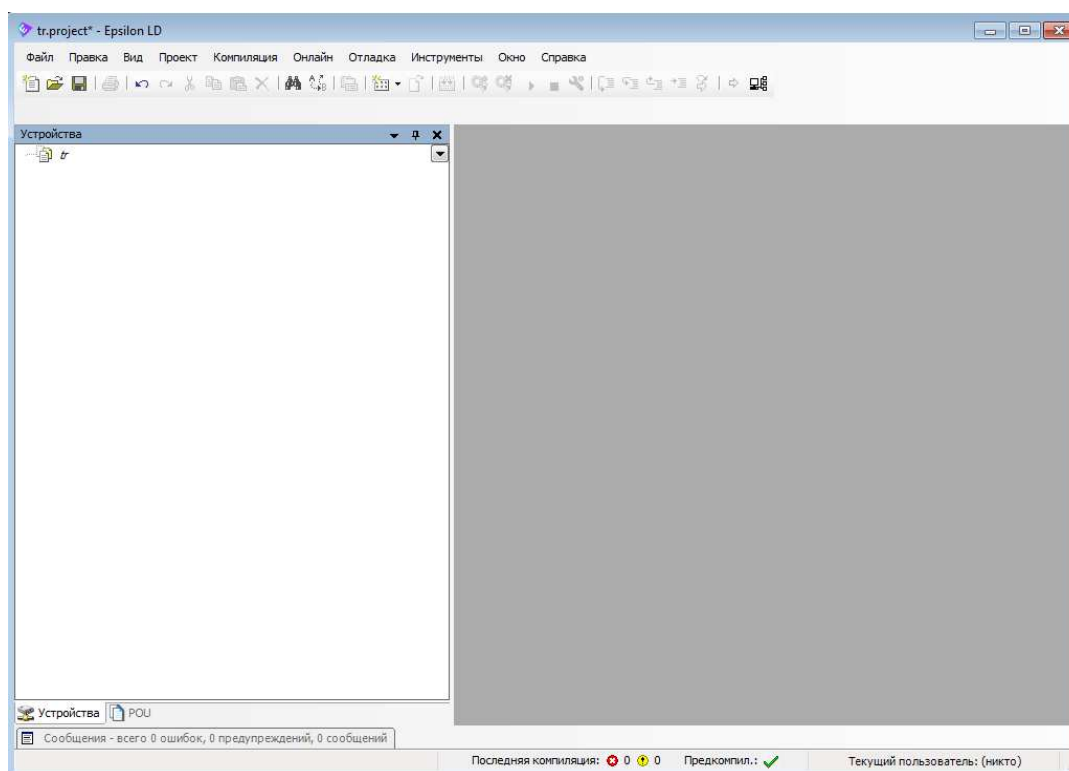
Для задания точек останова в онлайн режиме, откройте редактор нужного вам ROU, выделите требуемую строку и нажмите клавишу <F9> либо примените команду «Переключить точку останова» из меню «Онлайн». После этого точка останова появится на экране. После попадания программы в точку останова её можно будет выполнять по шагам с помощью клавиши <F8> или команды «Шаг детальный». При этом будут также выполняться шаги экземпляра функционального блока. Чтобы пропустить шаги функционального блока,

используйте клавишу <F10> или команду «Шаг поверху». При этом текущие значения всех переменных на ПЛК будут отображаться на текущей позиции выполнения программы.

С помощью команды «Точки останова» из меню «Вид» можно открыть «Диалог точек останова». Здесь можно просмотреть и изменить текущие точки останова, а также задать новые.

КОНФИГУРИРОВАНИЕ АППАРАТНОЙ ЧАСТИ КОНТРОЛЛЕРА REGUL R600

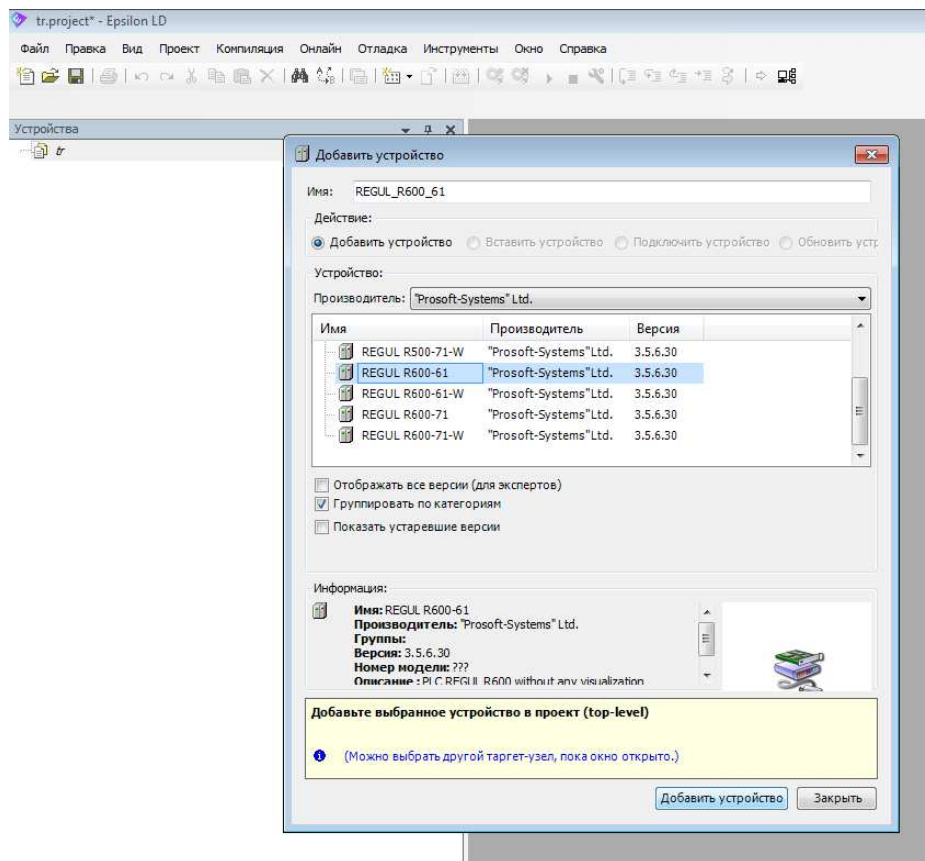
Для того чтобы сконфигурировать аппаратную часть контроллера Regul R600, необходимо запустить среду разработки и создать новый или открыть существующий проект. После открытия нового проекта в среде разработки, в левой части экрана, появится проводник устройств.



Проводник устройств

Один проект может быть использован для конфигурирования и программирования одного или нескольких контроллеров.

После загрузки проекта необходимо добавить новое устройство «Regul R600». Для этого следует нажать правой кнопки мыши по корневому узлу  в дереве устройств и выбрать пункт меню «Добавить устройство» (Add Device). Откроется форма добавления новых устройств, в которой, левой кнопкой мыши, нужно выбрать устройство «Regul R600» и в нижней части формы нажать кнопку «Добавить устройство».



Форма добавления нового устройства

Существует 4 типа устройства Regul R600:

- Regul R600-61, контроллер Regul R600 без визуализации
- Regul R600-61-W, контроллер Regul R600 с поддержкой web-визуализации
- Regul R600-71, контроллер Regul R600 с VGA выходом
- Regul R600-71-W, контроллер Regul R600 с VGA выходом и возможностью web-визуализации

Конфигурирование структуры кольцевой шины

Контроллер REGUL R600 состоит из одного основного крейта, содержащего модуль центрального процессора (ЦП), и максимум 255 крейтов расширения.

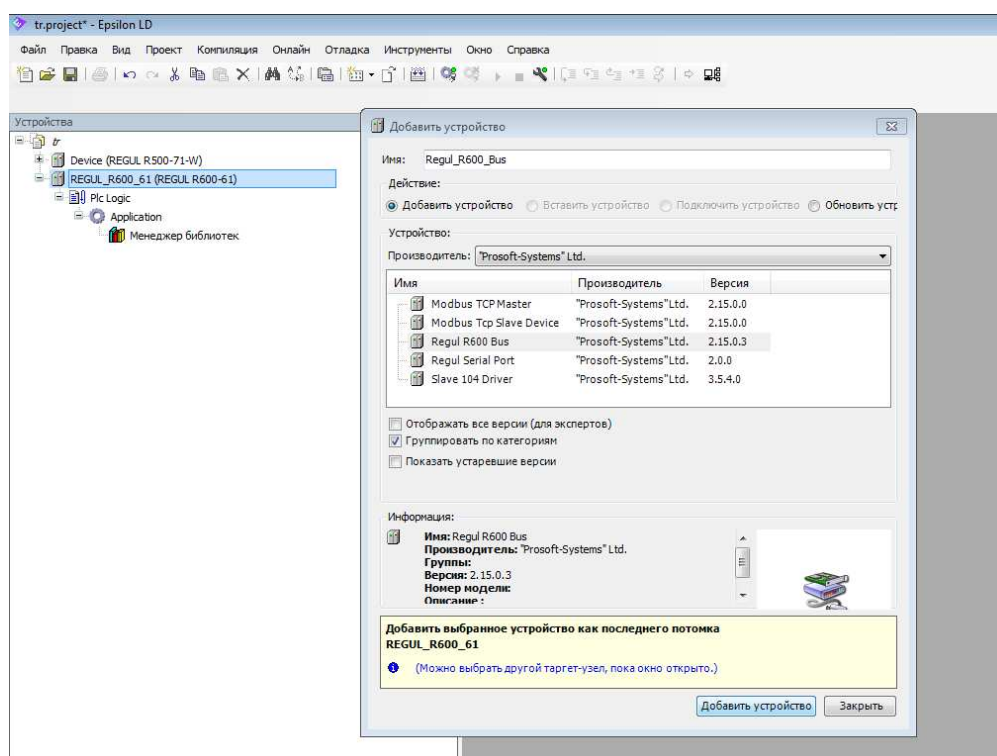
Кроме того, контроллер может применяться в резервированных и нерезервированных системах. При этом, в случае, когда контроллер применяется в резервированной системе, каждый крейт может быть резервирован полностью или частично.

В случае полного резервирования, одному крейту в проекте в действительности соответствует два крейта с одинаковым набором модулей и с одним и тем же адресом (см. далее).

При частичном резервировании резервируются только модули блока питания (БП) и, если крейт содержит модули ЦП, то модули ЦП. Таким образом, при частичном резервировании одному крейту в проекте в действительности соответствует только один крейт с двойным набором модулей БП и, если необходимо, модулей ЦП.

Для того чтобы редактировать структуру шины контроллера Regul R600, необходимо добавить в проект устройство «Шина расширения» (Regul R600 Bus).

Процедура добавления шины расширения аналогична добавлению устройства Regul R600: необходимо в дереве устройств щелкнуть правой кнопкой мыши по устройству Regul R600. В появившемся контекстном меню необходимо выбрать пункт «Добавить устройство» (Add Device). При этом появится форма добавления нового устройства. После выбора устройства «Regul R600 Bus» и нажатия на кнопку «Добавить устройство» (Add Device), в проект добавится «Шина расширения».

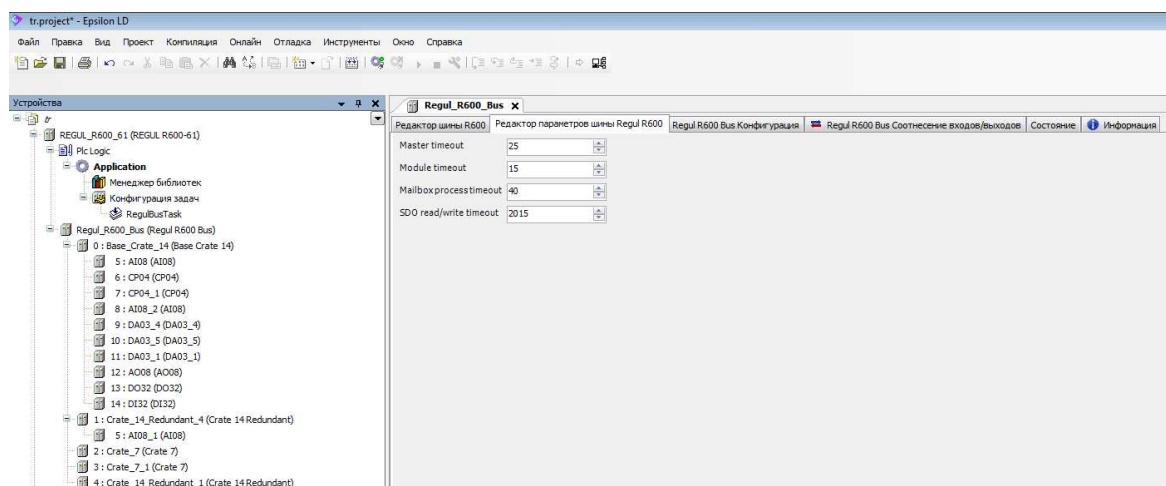


Форма добавления нового устройства (шина расширения)

Описание редактора шины

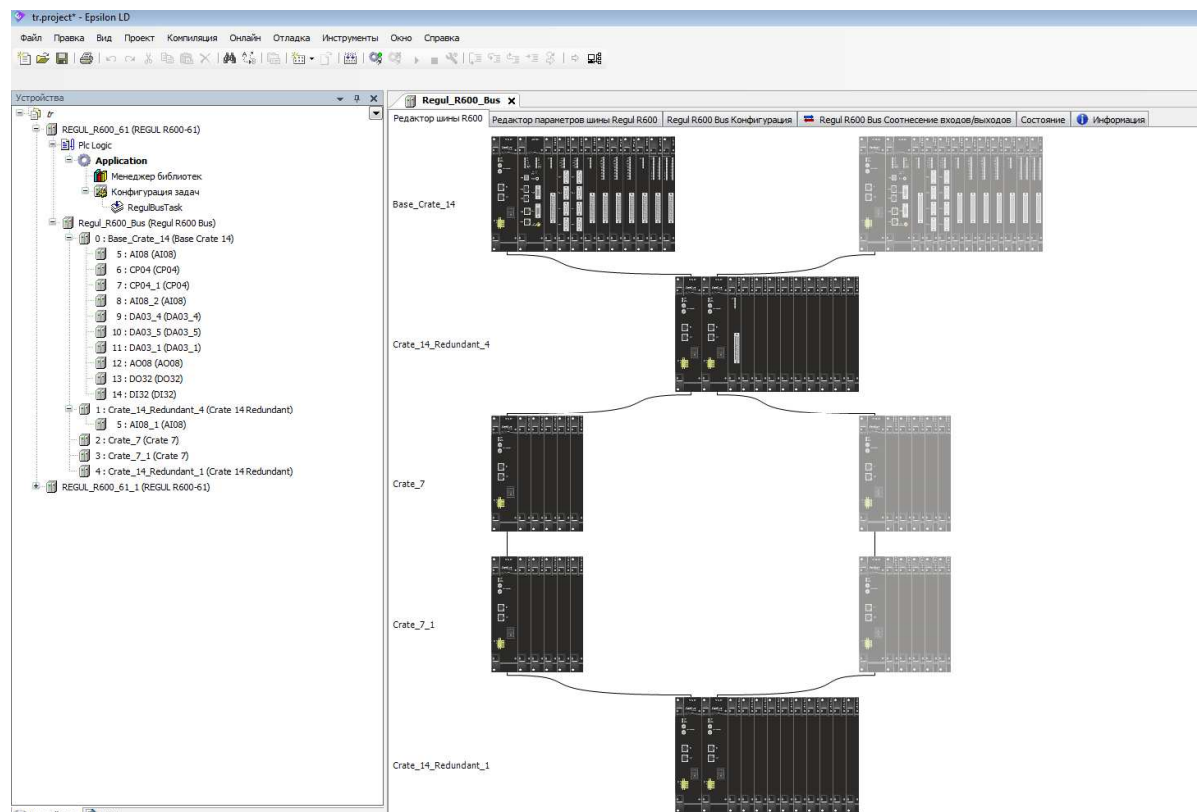
Для того чтобы открыть редактор, необходимо в дереве устройств, дважды щелкнуть левой кнопкой мыши по шине расширения и перейти на вкладку «Редактор параметров шины Regul R600». При помощи данного редактора можно задать такие параметры, как:

- Таймаут для перевода модуля в ошибку по условию "Нет пакетов EtherCAT от мастера" (Master timeout)
- Таймаут на не изменение хартбита модуля (Module timeout)
- Таймаут на обработку входящего майлбокса, по истечении которого майлбокс уничтожается (Mailbox process timeout)
- Таймаут на обработку SDO запроса (SDO read/write timeout)



Редактор параметров шины расширения

Также в окне данного редактора на вкладке «Редактор шины R600» представлена структура шины, на которой схематически изображены крейты и связи между ними. Двойным щелчком левой кнопки мыши по изображению крейта можно открыть его редактор.



Структура шины Regul R600

Конфигурирование крейтов

Крейт представляет собой набор модулей. Если в крейте есть модуль центрального процессора, то такой крейт является базовым, иначе - крейтом расширения. В шине расширения может быть только 1 базовый крейт, и его необходимо добавить первым. Также крейты различаются по количеству слотов (7 или 14) и по наличию резервных модулей питания (и центрального процессора, если крейт является базовым).

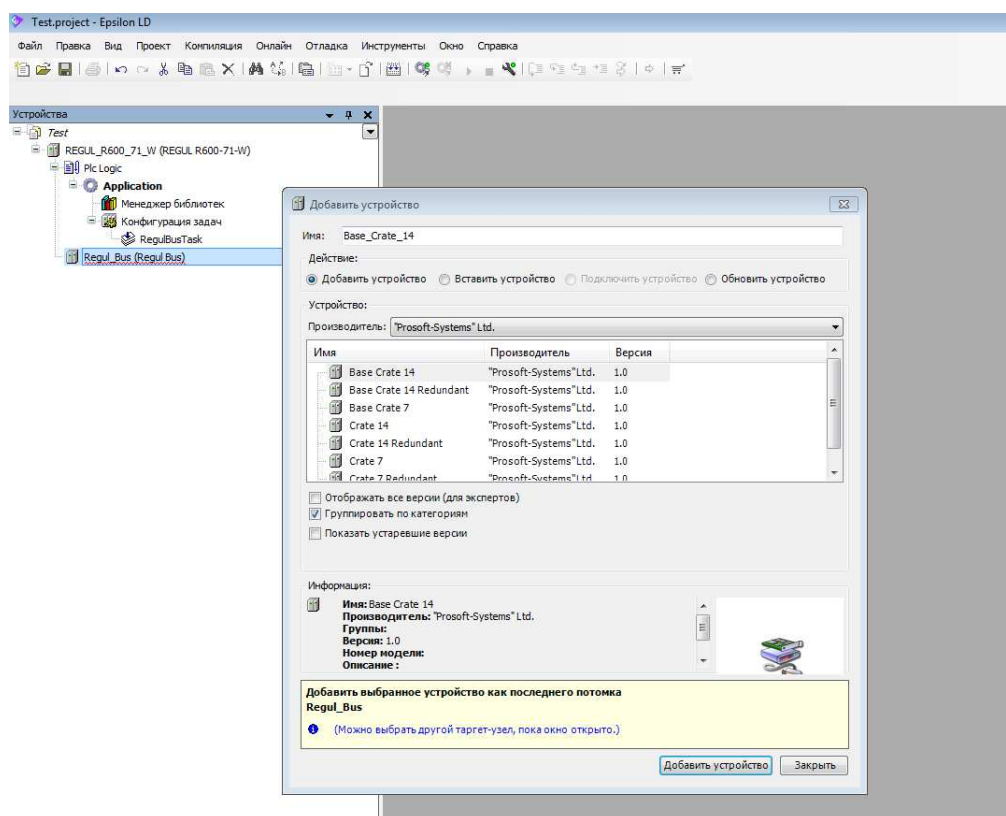
Добавление крейтов.

Для того чтобы добавить крейт, необходимо на панели «Устройства» щелкнуть правой кнопкой мыши по объекту «Шина расширения». В появившемся контекстном меню необходимо выбрать пункт «Добавить устройство» (Add Device). При этом появится форма добавления нового устройства. Возможны следующие варианты крейтов:

- Base Crate 14 (базовый крейт с 14 слотами)
- Base Crate 7 (базовый крейт с 7 слотами)
- Base Crate 14 Redundant (базовый резервированный крейт с 14 слотами)
- Crate 14 (крейт расширения с 14 слотами)

- Crate 14 Redundant (резервированный крейт расширения с 14 слотами)
- Crate 7 (крейт расширения с 7 слотами)
- Crate 7 Redundant (резервированный крейт расширения с 7 слотами)
- Crate Virtual (не участвует в формировании шины, является хранилищем для модулей ввода/вывода)

После выбора нужного типа крейта и нажатия на кнопку «Добавить устройство» (Add Device), в проект добавится новый крейт.



Форма добавления нового крейта

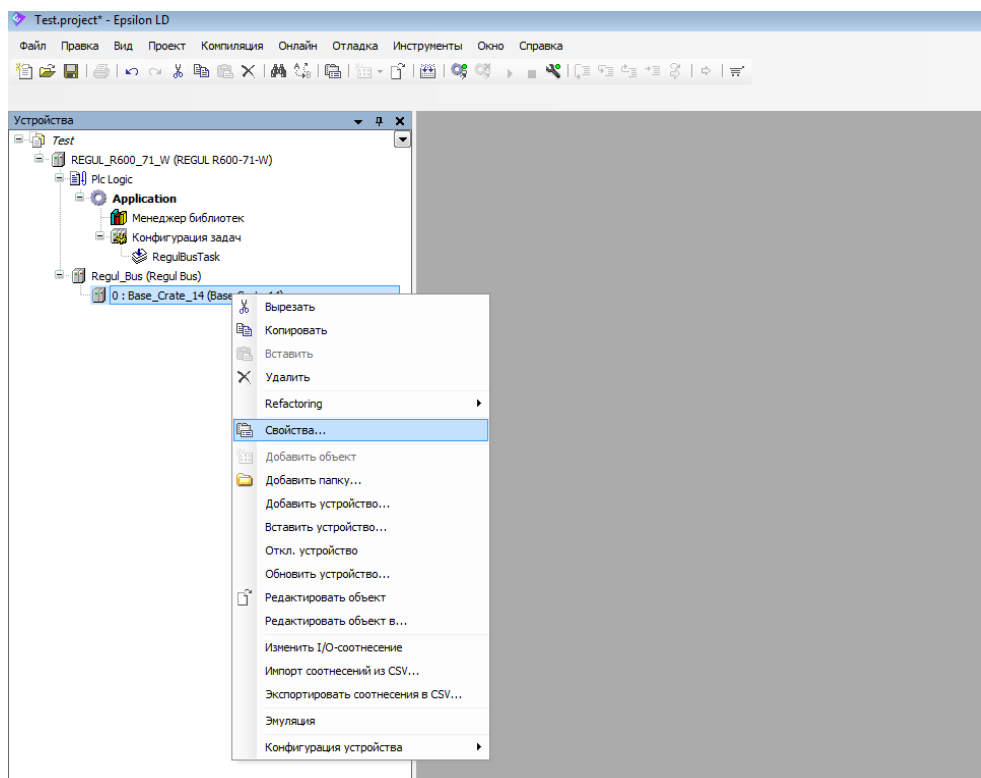
При добавлении первого крейта в шину, для него обязательно следует выбрать базовый тип (Base Crate 14, Base Crate 14 Redundant или Base Crate 7).

Если требуется сменить тип уже существующего базового крейта, то необходимо использовать команду контекстного меню «Обновить устройство», а не пытаться удалить данный базовый крейт и добавить заново новый крейт другого типа.

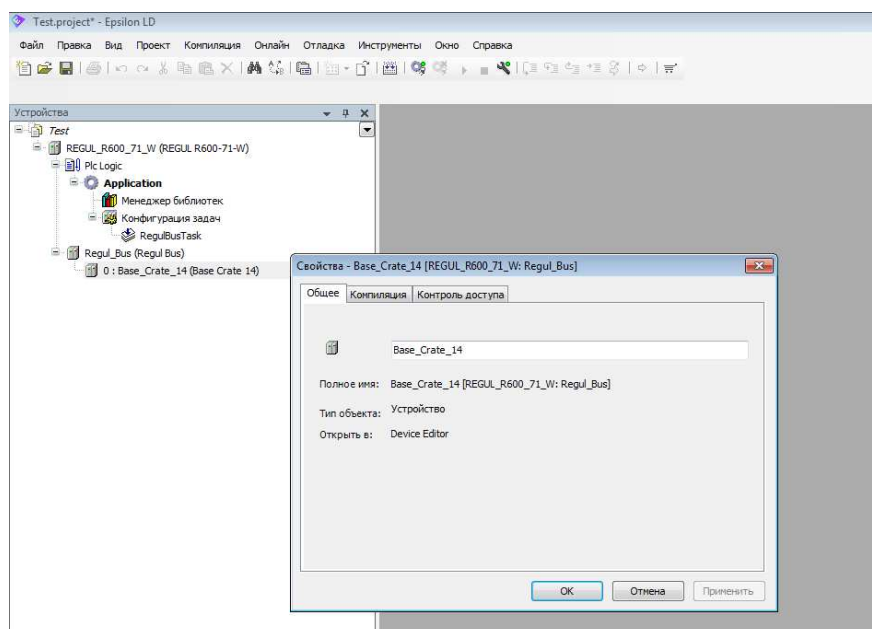
Переименование крейта расширения

Для того чтобы переименовать крейт, необходимо на панели «Устройство» щелкнуть правой кнопкой мыши по крейту. В появившемся контекстном меню необходимо выбрать

пункт «Свойства» (Properties). При этом появится диалоговое окно свойств крейта. Поле «Имя» позволяет задать новое имя для крейта. Аналогичным образом можно задать имя для любого устройства в проекте.



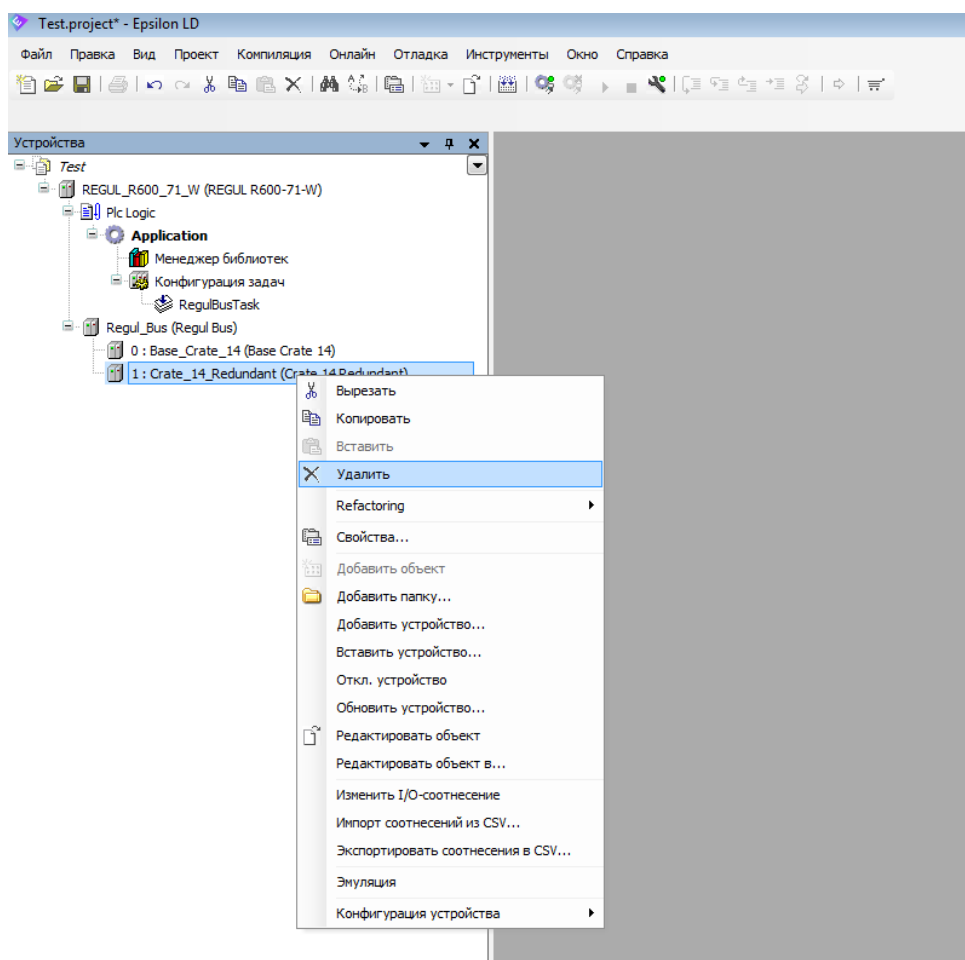
Свойства



Диалоговое окно свойств крейта

Удаление крейта расширения

Для того чтобы удалить крейт, необходимо на панели «Устройства» щелкнуть правой кнопкой мыши крейту. В появившемся контекстном меню необходимо выбрать пункт «Удалить» (Delete). Также можно выбрать крейт и нажать кнопку «Delete» на клавиатуре. Аналогичным образом можно удалить любые устройства в проекте.



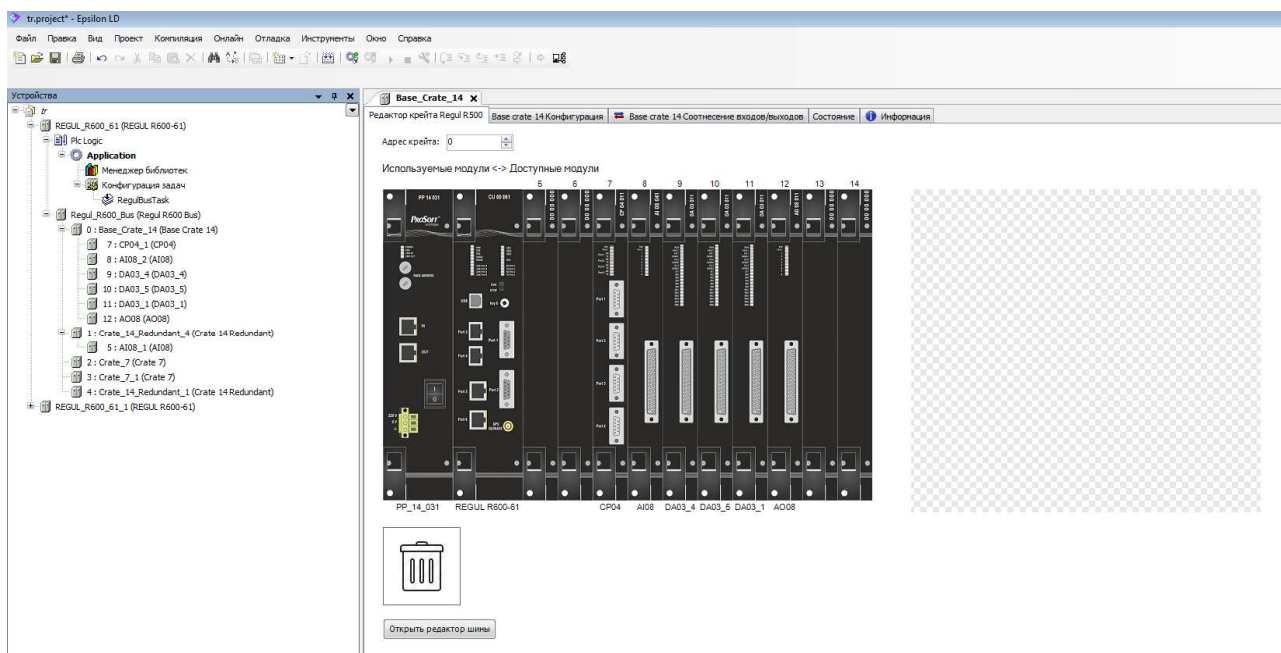
Удаление крейта

Редактор крейта

Для того чтобы открыть редактор какого-либо крейта, необходимо на панели «Устройства» (Devices) дважды щелкнуть левой кнопкой мыши по крейту.

Каждому крейту, входящему в состав контроллера REGUL R600 (в том числе и базовый крейту) сопоставляется уникальный адрес. Этот адрес задается двумя поворотными переключателями на лицевой панели модуля блока питания соответствующего крейта. Диапазон допустимых адресов от 0 до 255. Точно такой же адрес должен быть задан при конфигурировании крейта в среде разработки. Поле «Адрес крейта» в редакторе позволяет

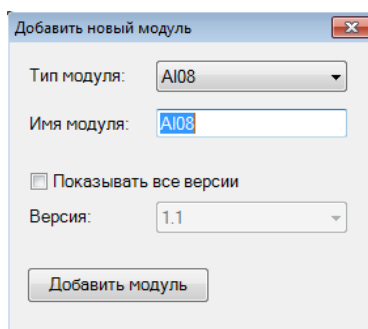
задать адрес выбранного крейта. Также адрес крейта отображается в дереве устройств перед именем крейта.



Редактор крейта

Основная часть редактора крейта состоит из двух панелей «Используемые модули» и «Доступные модули». На панели «Используемые модули» представлены все доступные для размещения слоты (посадочные места для модулей) и модули данного крейта, для которых задан номер слота (слоты CU/PP module зарезервированы для модулей питания и центрального процессора). Количество доступных слотов зависит от типа выбранного крейта.

Для добавления нового модуля в слот необходимо дважды щелкнуть левой кнопкой мыши по пустому слоту. Откроется диалоговое окно добавления нового модуля, где необходимо задать тип модуля при помощи выпадающего списка и его имя.



Диалоговое окно добавления нового модуля

Также новый модуль можно добавить стандартным способом, путем нажатия правой кнопкой мыши по крейту и выбора пункта меню «Добавить устройство» (Add Device). В

таком случае, для модуля не будет задан номер слота и он попадёт в список «Доступные модули». Для того чтобы задать номер слота для модуля, достаточно перетащить (drag and drop) его на свободный слот в списке «Установленные модули». Также можно перетаскивать уже установленные модули в область «Доступные модули» для того, чтобы освободить слот занимаемый модулем (может понадобиться при необходимости сменить слот уже установленного модуля). Кроме того, можно перетаскивать установленные модули между слотами. Занимаемый модулем слот отображается в дереве устройств перед именем данного модуля. Для удаления модуля перетащите его на изображение корзины.

При помощи кнопки «Открыть редактор шины» можно перейти к редактору шины.

Конфигурирование модулей

Добавление нового модуля

Процедура добавления новых модулей описана в главе «Редактор крейта».

Редактор модуля

Для того чтобы открыть редактор, необходимо на панели «Устройства» дважды щелкнуть левой кнопкой мыши по выбранному модулю.

Все редакторы модулей имеют типовую структуру и состоят из следующих групп:

– базовые параметры, общие для всех модулей:

- Требуемая главная версия прошивки (Major version required)
- Текущая версия прошивки (Version current)
- Требуемый id устройства (Device ID required)
- Текущий id устройства (Device ID current)
- Минимально необходимая версия прошивки (Version at least)

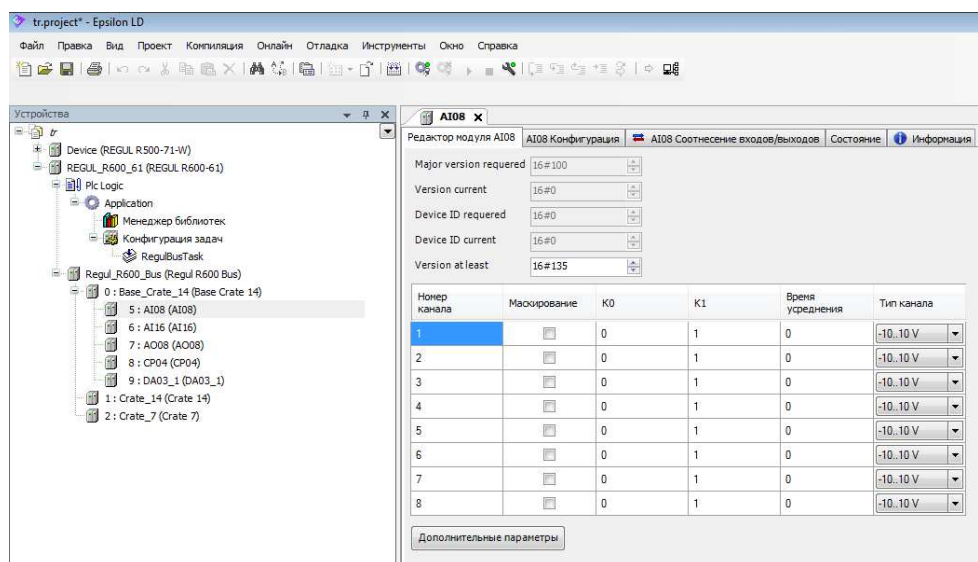
– параметры, специфичные для данного модуля (если такие есть)

– таблица параметров каналов с возможность чтения/записи значений параметров как в офлайн режиме, так и при подключении к контроллеру

– отдельной вкладки редактора, где осуществляется привязка переменных к входам/выходам модуля.

Задание параметров модуля.

Почти все модули контроллеров Regul R600 имеют ряд параметров, которые используются для того, чтобы настроить работу модуля под конкретную прикладную задачу. К таким параметрам, например, относятся маскирование каналов ввода/вывода, коэффициенты калибровки аналогового канала ввода/вывода, скорость обмена по каналам коммуникационных модулей и т.д.



Редактор модуля AI08

Конкретный список доступных для задания параметров модуля зависит от его типа (см. Приложение А).

Для изменения какого – либо параметра канала достаточно в столбце с нужным вам параметром модуля и в строке с требуемым номером канала изменить его значение. Если изменения были произведены в офлайн режиме, то новые параметры будут применены при следующей загрузке программы в ПЛК. Если изменения были произведены при подключении к контроллеру (онлайн режим), то новые параметры будут применены сразу. Следует отметить, что изменение параметров в онлайн режиме будет действительно до перезагрузки ПЛК или прикладной программы, в данном случае будут применены настройки указанные в проекте.

У некоторых редакторов модулей кроме основных параметров есть дополнительные, которые становятся доступны при нажатии на кнопку «Дополнительные параметры». Редактирование дополнительных параметров осуществляется так же, как и основных. Также в окне «Дополнительные параметры» присутствуют параметры, сохраняемые в ПЗУ модуля (калибровочные коэффициенты). Изменения значений таких параметров может

осуществляться только в онлайн-режиме (при подключении к контроллеру) и для сохранения новых значений необходимо нажать кнопку «Сохранить в ПЗУ».

Привязка входов/выходов

Вкладка «Соотнесение входов/выходов» служит для привязки входов и выходов модулей к переменным в проекте.

Далее под входами и выходами модулей понимаются как физические входы и выходы, так и логические входные или выходные значения, которые генерируются модулем в результате его работы или передаются ему из прикладной программы.

Список входов и выходов конкретного модуля, доступных для привязки, зависит от его типа (см. Приложение Б).

При соотнесении переменных и входов/выходов необходимо учитывать следующее:

- К переменным, требующим входа, нельзя обращаться посредством записи;
- Существующая переменная может быть соотнесена только с одним входом;
- Распределение памяти структур определяется целевой платформой;
- Тип канала обозначается в столбце «Переменная» иконкой  для входа и  для выхода.

В зависимости от устройства может быть показана следующая информация:

- Канал: символьное имя входа или выхода канала устройства;
- Адрес: адрес канала, например, "%QX3.0". Значение адреса можно изменить вручную. Это может потребоваться для изменения адресации в соответствии с данной аппаратной конфигурацией или сохранения значения адреса даже при изменении порядка модулей (по умолчанию это приводит к автоматическому изменению значений адреса). Учтите, что независимо от описания устройства вы можете изменить адрес только всего входа или выхода целиком, а не отдельных его подэлементов (бит-каналов). Таким образом, если вход или выход представлен в таблице соотнесений поддеревом, адрес можно изменить только у самого верхнего объекта (см. пример ниже: открыть можно только поле адреса в самой верхней строке). Чтобы зафиксировать значение адреса выделите его и нажмите <пробел>. При этом откроется поле ввода. Затем измените значение или оставьте как есть и закройте поле клавишей <Return>;

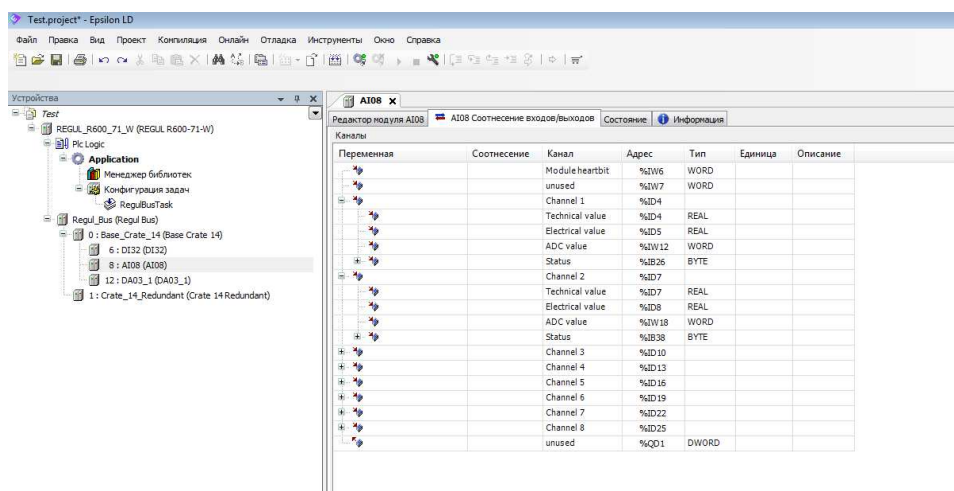
- Тип: тип данных входного или выходного канала, например, "BOOL". Если используется не стандартный тип данных, а структура или битовое поле, заданное в описании устройства, то он будет показан, только если включен в стандарт МЭК и в описании устройства помечен как тип МЭК. В противном случае поле таблицы будет пустым;
- Единица: единица значения параметра, например, "ms" для миллисекунд;
- Описание: краткое описание параметра;
- Текущее значение параметра: текущее значение параметра, показывается в онлайн-режиме;

Для того чтобы привязать переменную ко входу или выходу модуля, необходимо в списке входов/выходов дважды щелкнуть по ячейке в столбце «Переменная» в строке, соответствующей необходимому входу или выходу. Далее необходимо ввести имя новой переменной или нажать на появившуюся в ячейке кнопку , чтобы выбрать уже существующую в переменной.

Двойной щелчок левой кнопкой мыши по ячейке в столбце «Описание» в строке какого-либо входа/выхода позволяет задать описание этого входа/выхода.

Чтобы сбросить все привязки переменных к входам/выходам модуля необходимо нажать на кнопку «Сброс сопоставления».

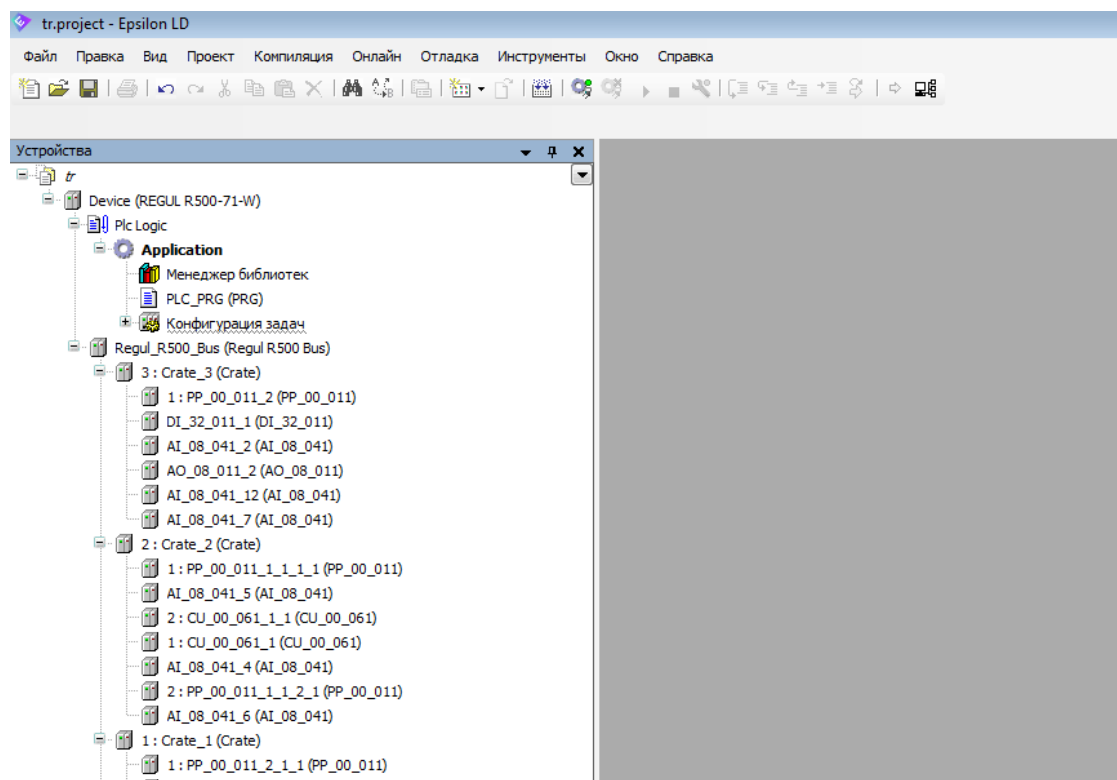
Если опция «Всегда обновлять переменные» активирована, то все переменные будут обновляться в каждом цикле задачи цикла шины, независимо от того, используются ли они обычно или сопоставлены с входом или выходом.



Сопоставление входов/выходов

КОНФИГУРИРОВАНИЕ АППАРАТНОЙ ЧАСТИ КОНТРОЛЛЕРА REGUL R500

Для того чтобы сконфигурировать аппаратную часть контроллера Regul R500, необходимо запустить среду разработки и создать новый или открыть существующий проект. После открытия файла проекта в среде разработки в левой части экрана появится проводник устройств.



Проводник устройств

Один проект может быть использован для конфигурирования и программирования одного или нескольких контроллеров.

После загрузки проекта необходимо добавить новое устройство «Regul R500». Для этого следует нажать правой кнопки мыши по корневому узлу  в дереве устройств и выбрать пункт меню «Добавить устройство» (Add Device). Откроется форма добавления новых устройств, в которой, левой кнопкой мыши, нужно выбрать устройство «Regul R500» и в нижней части формы нажать кнопку «Добавить устройство».

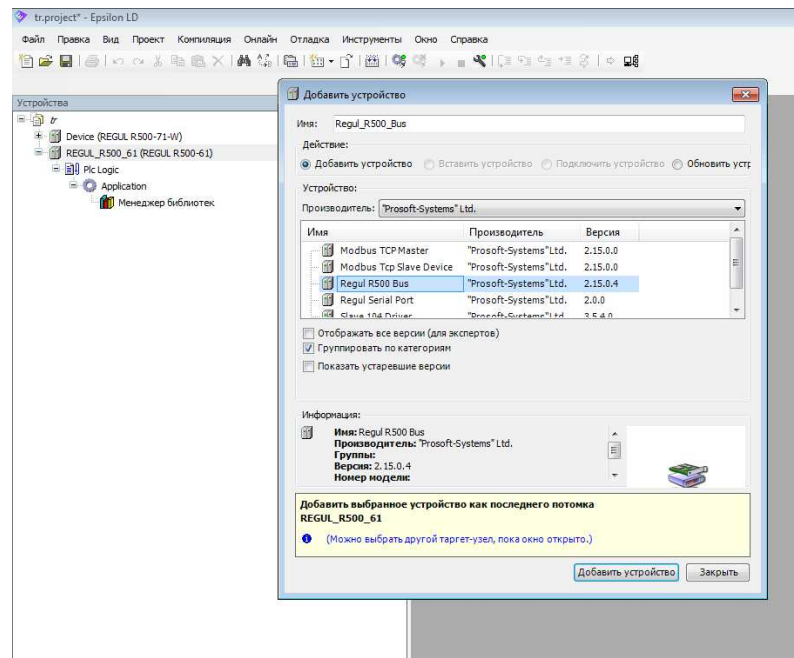
Конфигурирование структуры кольцевой шины

Контроллер Regul R500 имеет блочно-модульную конструкцию, состоящую из одного или нескольких крейтов, которые включают в себя модули различного типа и шасси, закрепленные на несущую рейку. Количество модулей разного типа (исключая оконечные модули) в составе одного крейта не должно превышать 25 штук.

При программной конфигурации шины контроллера необходимо придерживаться следующих правил:

- Во всей шине может быть только один либо два модуля центрального процессора, в случае наличия двух модулей ЦП они должны работать на разных шинах (номер шины задаётся в редакторе модуля центрального процессора). При необходимости резервирования крейта с модулем центрального процессора, данный модуль должен быть настроен на работу в режиме «любая шина» (MB*).
- Если в контроллере используется один модуль центрального процессора, то каждый крейт должен содержать один блок питания настроенный на работу с соответствующей шиной.
- Если в контроллере задействованы обе шины, то каждый нерезервированный крейт должен содержать 2 блока питания, которые настроены на работы с обеими шинами соответственно. Если крейт резервирован, то он должен иметь 1 блок питания, работающий в режиме «любая шина» (MB*).
- В случае полного резервирования, одному крейту в проекте в действительности соответствует два крейта с одинаковым набором модулей и с одним и тем же адресом.

Для того чтобы редактировать структуру шины контроллера Regul R500, необходимо добавить в проект устройство «Шина расширения» (Regul R500 Bus). В дереве устройств следует щелкнуть правой кнопкой мыши по устройству Regul R500. В появившемся контекстном меню необходимо выбрать пункт «Добавить устройство» (Add Device). При этом появится форма добавления нового устройства. После выбора устройства «Regul R500 Bus» и нажатия на кнопку «Добавить устройство» (Add Device), в проект добавится «Шина расширения».

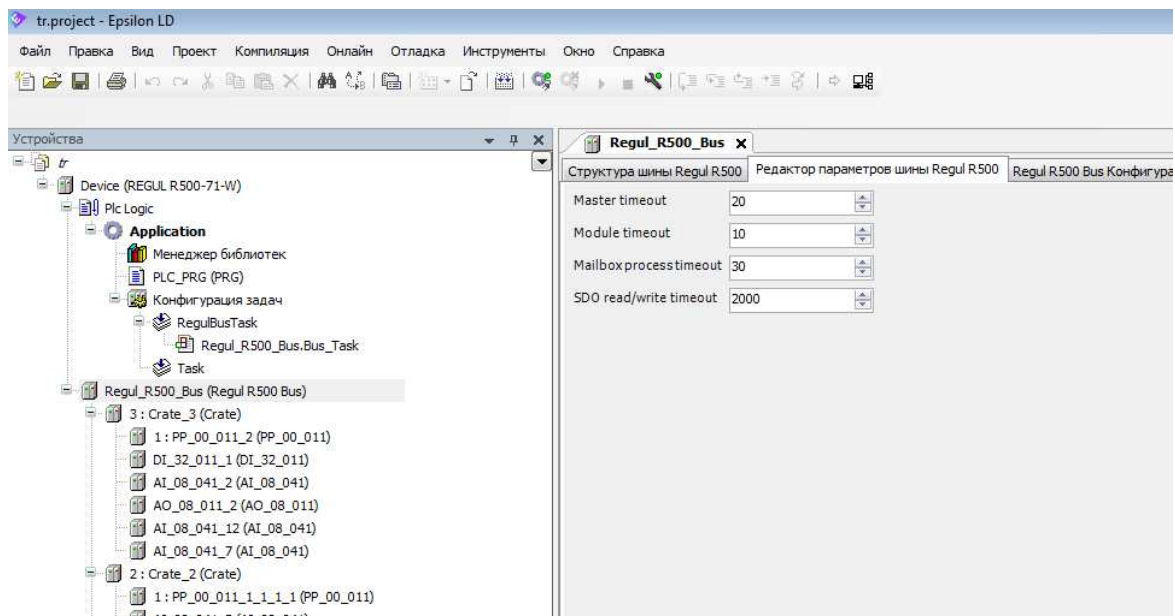


Форма добавления нового устройства (шина расширения)

Описание редактора шины

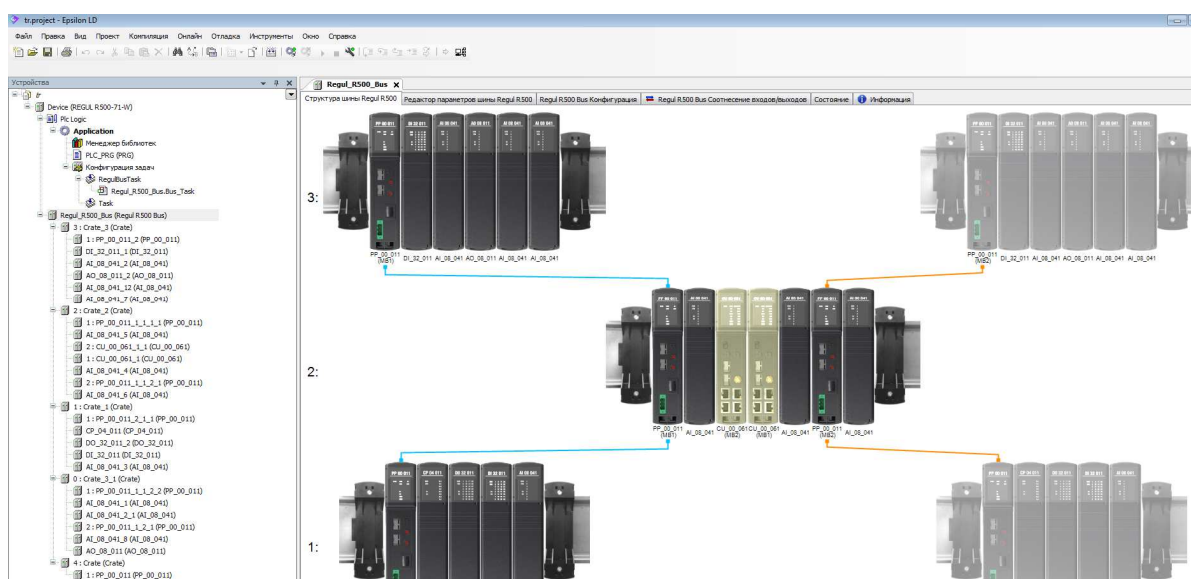
Для того чтобы открыть редактор шины, необходимо в дереве устройств, дважды щелкнуть левой кнопкой мыши по устройству «Regul R500 Bus» и перейти на вкладку «Редактор параметров шины Regul R500». При помощи данного редактора можно задать такие параметры, как:

- Таймаут для перевода модуля в ошибку по условию "Нет пакетов EtherCAT от мастера" (Master timeout)
- Таймаут на не изменение хартбита модуля (Module timeout)
- Таймаут на обработку входящего майлбокса, по истечении которого майлбокс уничтожается (Mailbox process timeout)
- Таймаут на обработку SDO запроса (SDO read/write timeout)



Редактор параметров шины расширения R500

Также в окне данного редактора на вкладке «Структура шины Regul R500» представлена структура шины, на которой схематически изображены крейты и связи между ними. Двойным щелчком левой кнопки мыши по изображению крейта можно открыть его редактор.



Структура шины Regul R500

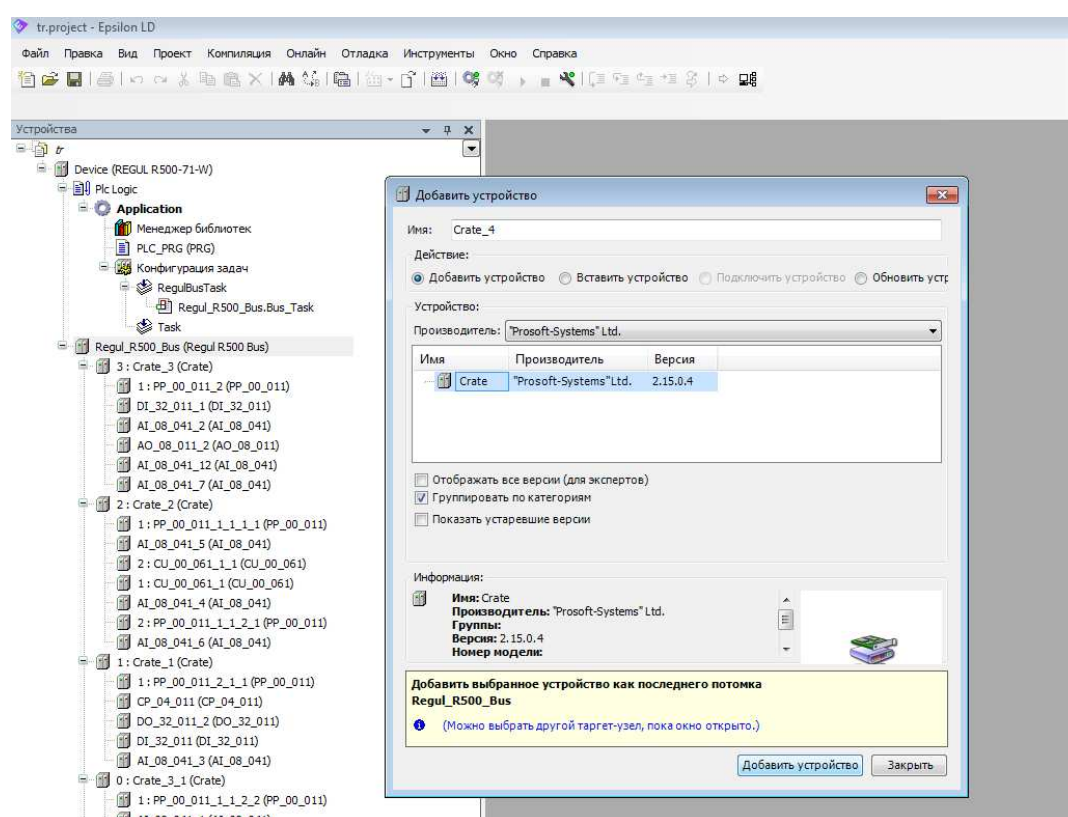
Серым цветом отображаются резервированные крейты, которые нельзя редактировать непосредственно.

Конфигурирование крейтов

Крейт представляет собой набор различных модулей контроллера Regul R500. К шине расширения можно подключить максимум 255 крейтов.

Добавление крейтов.

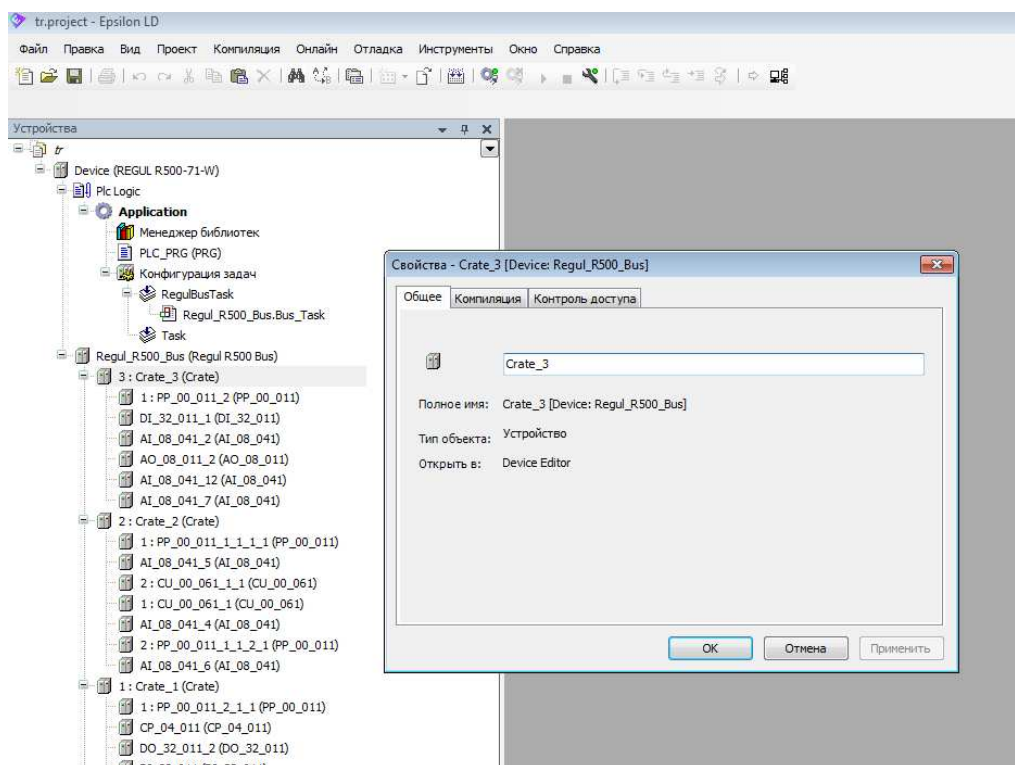
Для того чтобы добавить крейт, необходимо на панели «Устройства» щелкнуть правой кнопкой мыши по объекту «Regul R500 Bus». В появившемся контекстном меню необходимо выбрать пункт «Добавить устройство» (Add Device). При этом появится форма добавления нового устройства. После выбора устройства «Crate» и нажатия на кнопку «Добавить устройство» (Add Device), в проект добавится новый крейт.



Форма добавления нового крейта

Переименование крейта расширения

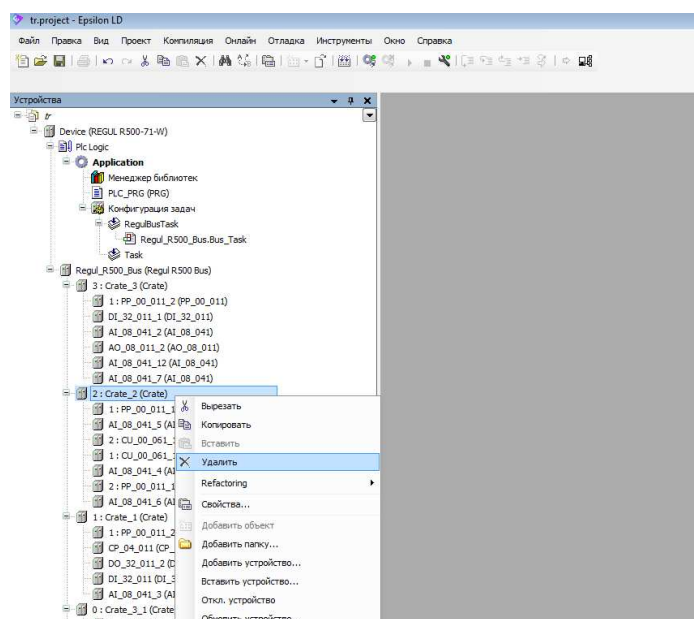
Для того чтобы переименовать крейт, необходимо на панели «Устройство» щелкнуть правой кнопкой мыши по устройству «Crate». В появившемся контекстном меню необходимо выбрать пункт «Свойства» (Properties). При этом появится диалоговое окно свойств крейта. Поле «Имя» позволяет задать новое имя для крейта. Аналогичным образом можно задать имя для любого устройства в проекте.



Свойства

Удаление крейта расширения

Для того чтобы удалить крейт, необходимо на панели «Устройства» щелкнуть правой кнопкой мыши крейту. В появившемся контекстном меню необходимо выбрать пункт «Удалить» (Delete). Также можно выбрать крейт и нажать кнопку «Delete» на клавиатуре. Аналогичным образом можно удалить любые устройства в проекте.

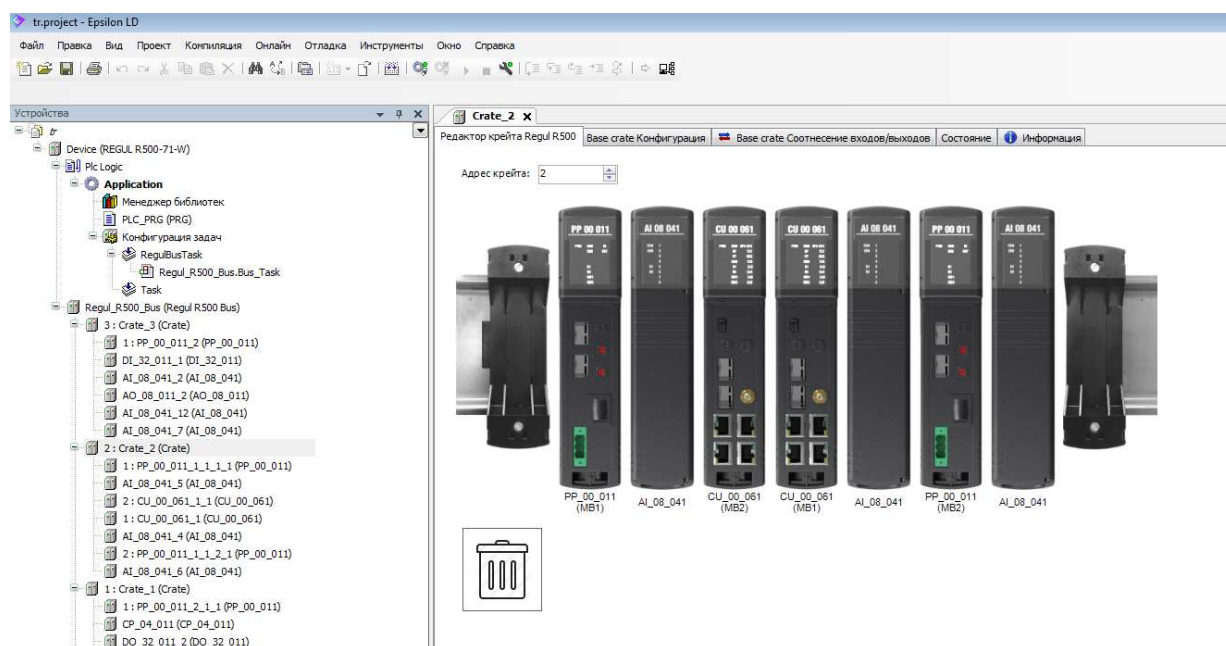


Удаление крейта

Редактор крейта

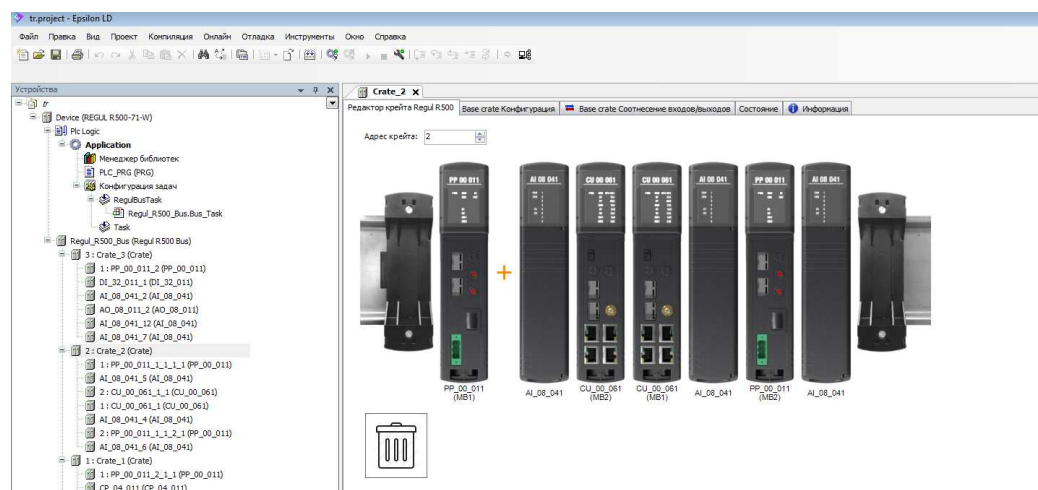
Для того чтобы открыть редактор какого-либо крейта, необходимо на панели «Устройства» (Devices) дважды щелкнуть левой кнопкой мыши по крейту.

Каждому крейту, входящему в состав контроллера REGUL R500 сопоставляется уникальный адрес. Этот адрес задается двумя поворотными переключателями на лицевой панели модуля блока питания соответствующего крейта. Диапазон допустимых адресов от 0 до 255. Точно такой же адрес должен быть задан при конфигурировании крейта в среде разработки. Поле «Адрес крейта» в редакторе позволяет задать адрес выбранного крейта. Также адрес крейта отображается в дереве устройств перед именем крейта.



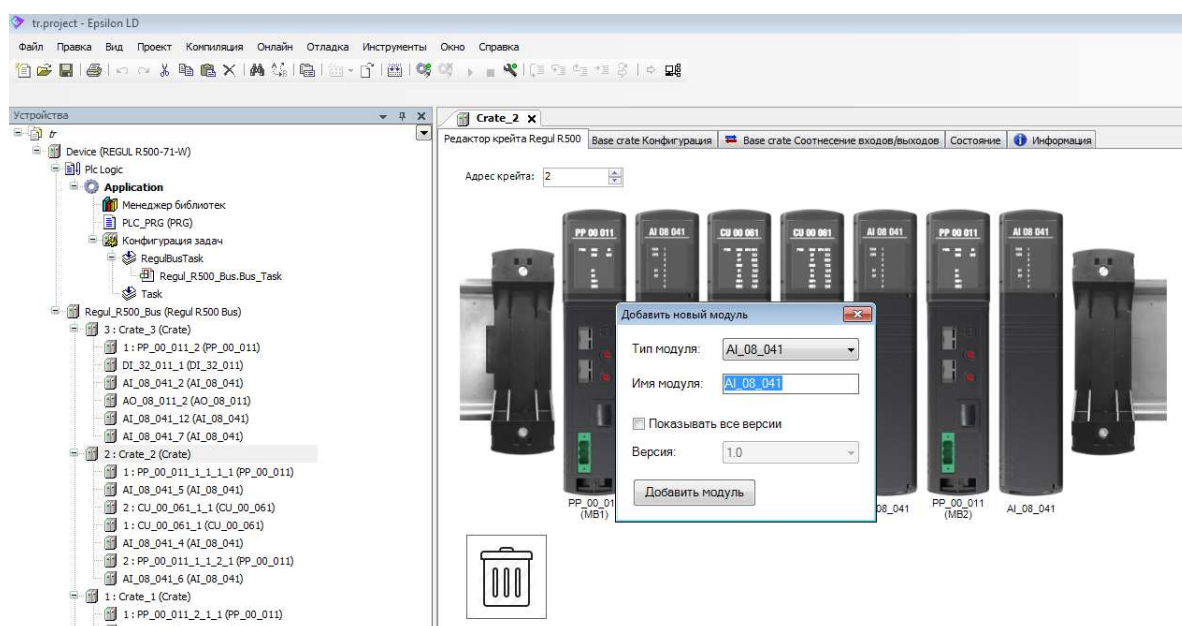
Редактор крейта

Основная часть редактора крейта состоит из панели с изображениями модулей. Порядок следования модулей в проекте должен совпадать с физическим порядком размещения модулей в крейте. Для добавления нового модуля в проект следует навести мышь на свободное пространство между изображениями модулей в редакторе, после этого в данном месте появиться знак «+».



Добавление нового модуля

При нажатии на знак «+» появится диалоговое окно добавления нового модуля. В данном окне необходимо выбрать тип модуля из выпадающего списка и задать имя. При необходимости, можно указать версию устройства (по умолчанию используется новейшая версия для данного устройства). Также новые модули можно добавлять через дерево устройств.



Диалоговое окно добавления нового модуля

Для того чтобы переместить существующий модуль в другое место, необходимо перетащить его изображение в редакторе (drag and drop). Изображения модулей допускается перетаскивать в свободное пространство между другими модулями, при этом, также как и при добавлении модуля, должен появиться знак «+». Для удаления модуля из проекта

необходимо перетащить его в область, обозначенную изображением корзины. Двойным кликом мыши по изображению модуля можно открыть его редактор.

Конфигурирование модулей

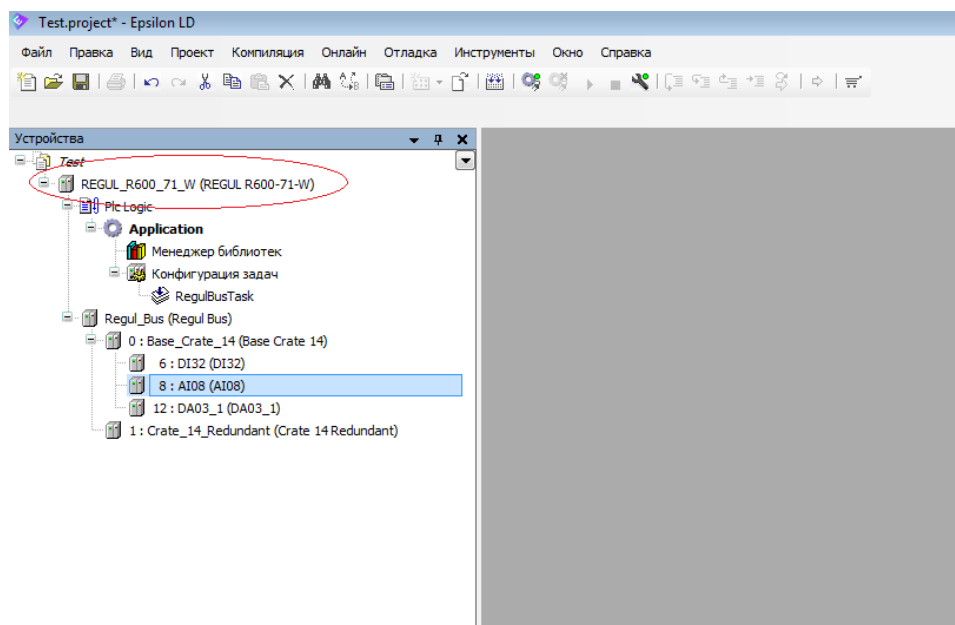
Конфигурирование модулей осуществляется так же, как и для контроллера Regul R600. Единственное отличие в том, что для контроллера Regul R500 необходимо непосредственно добавлять модули центрального процессора и модули питания в крейты. Также для данных модулей необходимо задавать номер шины. Для модуля центрального процессора возможны три варианта: работа по первой шине (MB1), работа по второй шине (MB2) и любая шина (MB*). Для модуля питания можно выбрать один из следующих режимов:

- работа по первой шине (MB1),
- работа по второй шине (MB2),
- любая шина (MB*),
- режим, когда модуль не участвует в формировании шины, а обеспечивает только питания для остальных модулей в крейте (None).

ОБЗОР ДОПОЛНИТЕЛЬНЫХ ИНСТРУМЕНТОВ

Сканер сети

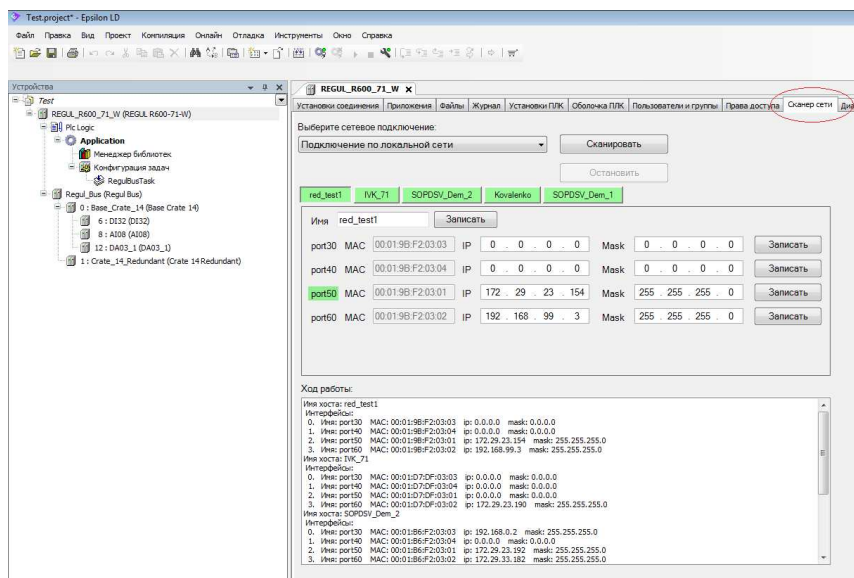
Сканер сети позволяет найти все подключенные к компьютеру ПЛК, задать для них сетевое имя, IP – адрес и маску подсети. Для того чтобы воспользоваться сканером сети откройте редактор устройства, двойным кликом левой кнопки мыши по любому устройству типа Regul в дереве объектов.



Устройство Regul R600

После открытия редактора перейдите на вкладку «Сканер сети». На данной вкладке выберите своё сетевое подключение и нажмите кнопку «Сканировать». Будет выведен список всех ПЛК Regul, найденных в вашей сети. Новые контроллеры будут выделены зелёным цветом.

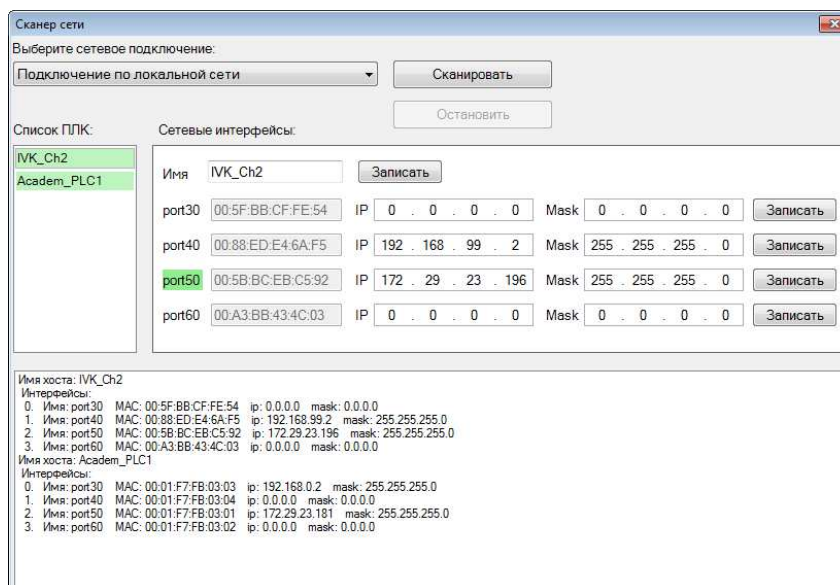
Нажмите на вкладку с именем ПЛК, для которого требуется узнать или задать новый IP – адрес. Вам будет представлена таблица со всеми сетевыми интерфейсами выбранного контроллера. Зеленым цветом будет выделен порт, с которого контроллер отвечает на сетевые запросы. Для того чтобы поменять IP-адрес контроллера введите новый IP – адрес и маску подсети в поле IP и Mask соответственно, затем нажмите кнопку «Записать».



Сканер сети

Также при помощи сканера сети можно изменить сетевое имя ПЛК, для этого запишите новое имя контроллера в поле «Имя» и нажмите кнопку «Записать».

Начиная с версии 1.5 пакета расширения Regul, сканер сети располагается в меню «Инструменты». Функционал остался прежним.



Диагностика ПЛК

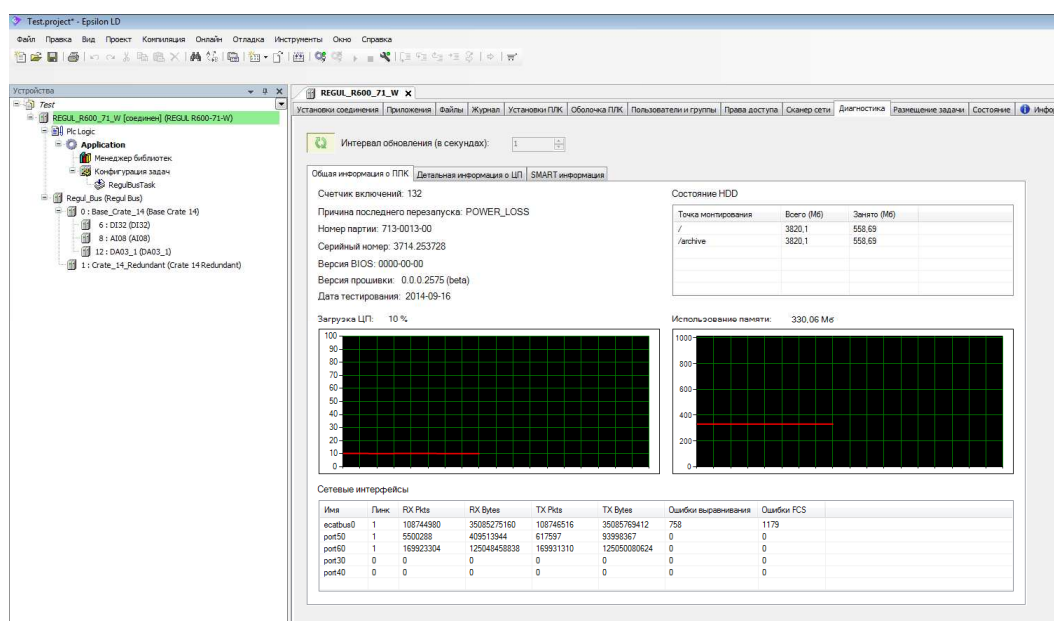
Вкладка «Диагностика» редактора устройства Regul позволяет узнать различную информацию о состоянии ПЛК в реальном времени. Для запуска диагностики установите интервал обновления информации в секундах и нажмите кнопку обновления.

Диагностическая информация делится на 3 группы:

- общая информация о ПЛК;
- детальная информация о ЦП (центральном процессоре);
- SMART информация.

Общая информация о ПЛК содержит следующие параметры:

- счётчик включений;
- причина последнего перезапуска;
- номер партии;
- серийный номер;
- версия BIOS;
- версия прошивки;
- дата тестирования;
- загрузка ЦП и использование оперативной памяти
- сетевые интерфейсы
- состояние HDD

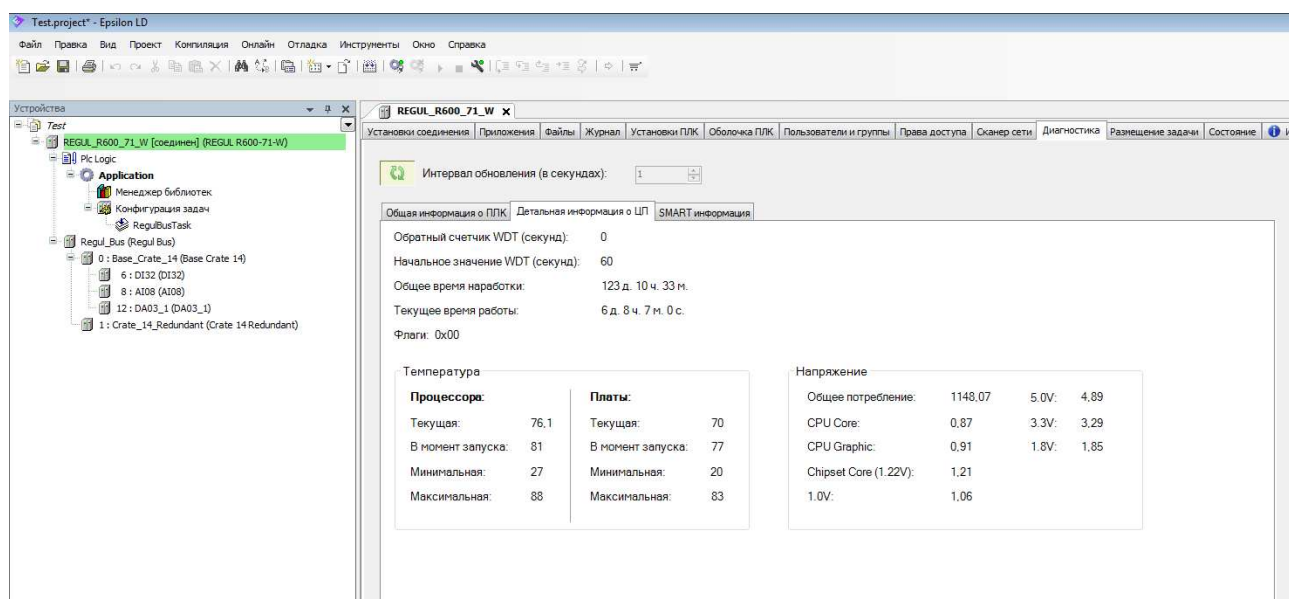


Диагностика (Общая информация)

Вкладка «Детальная информация о ЦП» позволяет узнать:

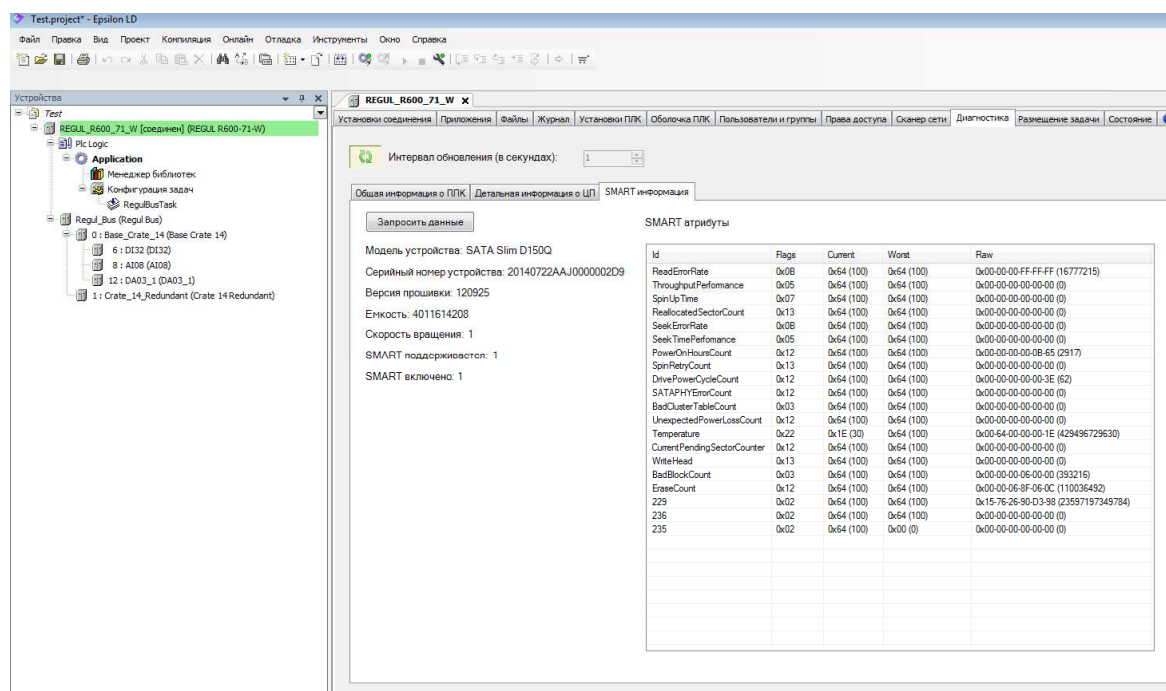
- начальное значение WDT;
- обратный счётчик WDT;
- общее время наработки;
- текущее время работы;

- температуру процессора и платы;
- потребляемое напряжение;



Диагностика (Детальная информация о ЦП)

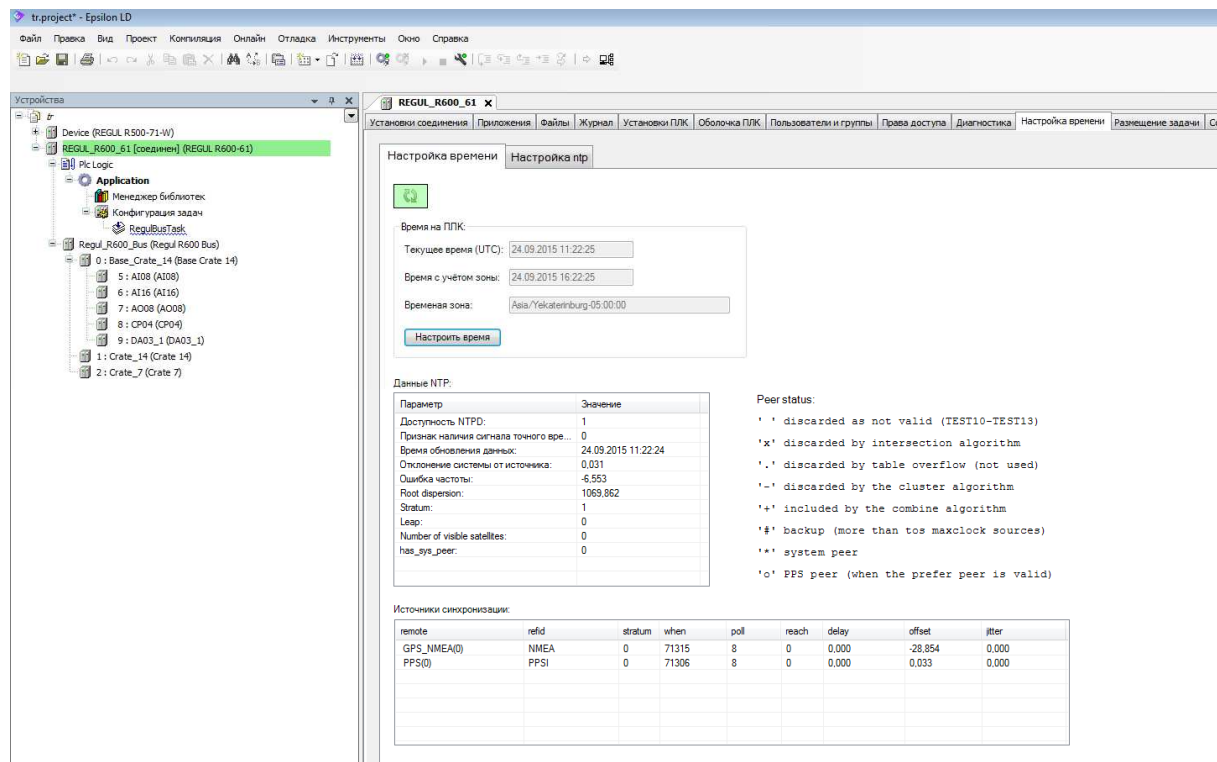
Последняя вкладка «SMART информация» содержит данные SMART о состоянии жёсткого диска.



Диагностика (SMART информация)

Настройка времени

Вкладка «Настройка времени» редактора устройства Regul позволяет узнать и задать текущее время на контроллере, а также изменить настройки ntp.

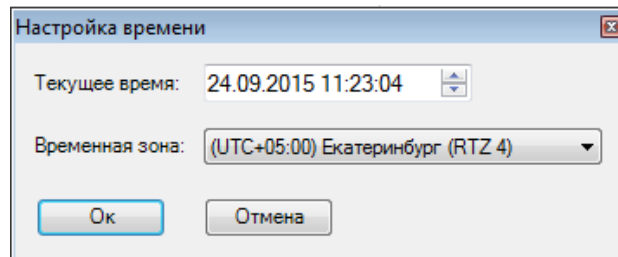


Настройка времени

Для того чтобы узнать или настроить текущее время на ПЛК установите с ним соединение и нажмите кнопку обновления. На экран будет выведена следующая информация:

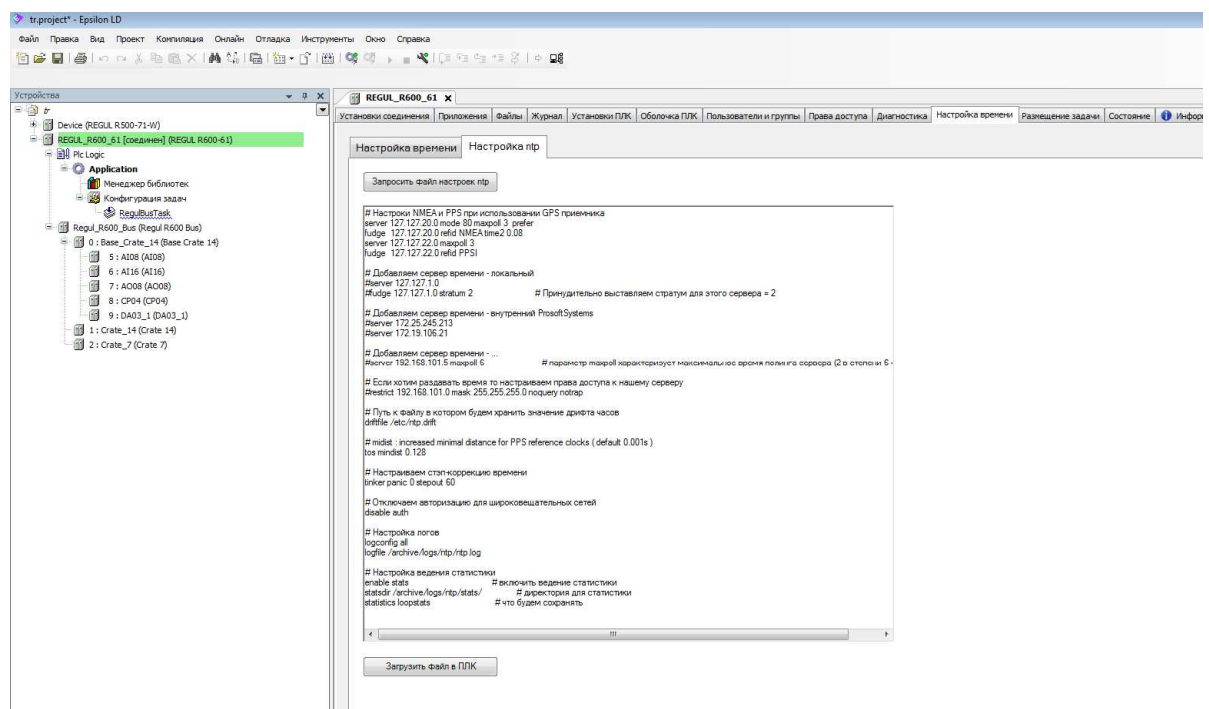
- Текущее время в UTC
- Временная зона
- Текущее время с учётом временной зоны
- Данные NTP
- Информация по источникам синхронизации времени

Чтобы настроить текущее время или временную зону нажмите на кнопку «Настроить время», появится диалоговое окно. После выбора новой временной зоны или смены времени нажмите кнопку «Ок» для сохранения новых значений в контроллер.



Диалоговое окно настройки времени

Для настройки ntp перейдите на вкладку «Настройка ntp» и нажмите на кнопку «Запросить файл настроек». Файл настроек будет загружен с контроллера и подставлен в текстовое поле для редактирования. После внесения необходимых изменений нажмите на кнопку «Загрузить файл в ПЛК» для сохранения нового файла в контроллер.



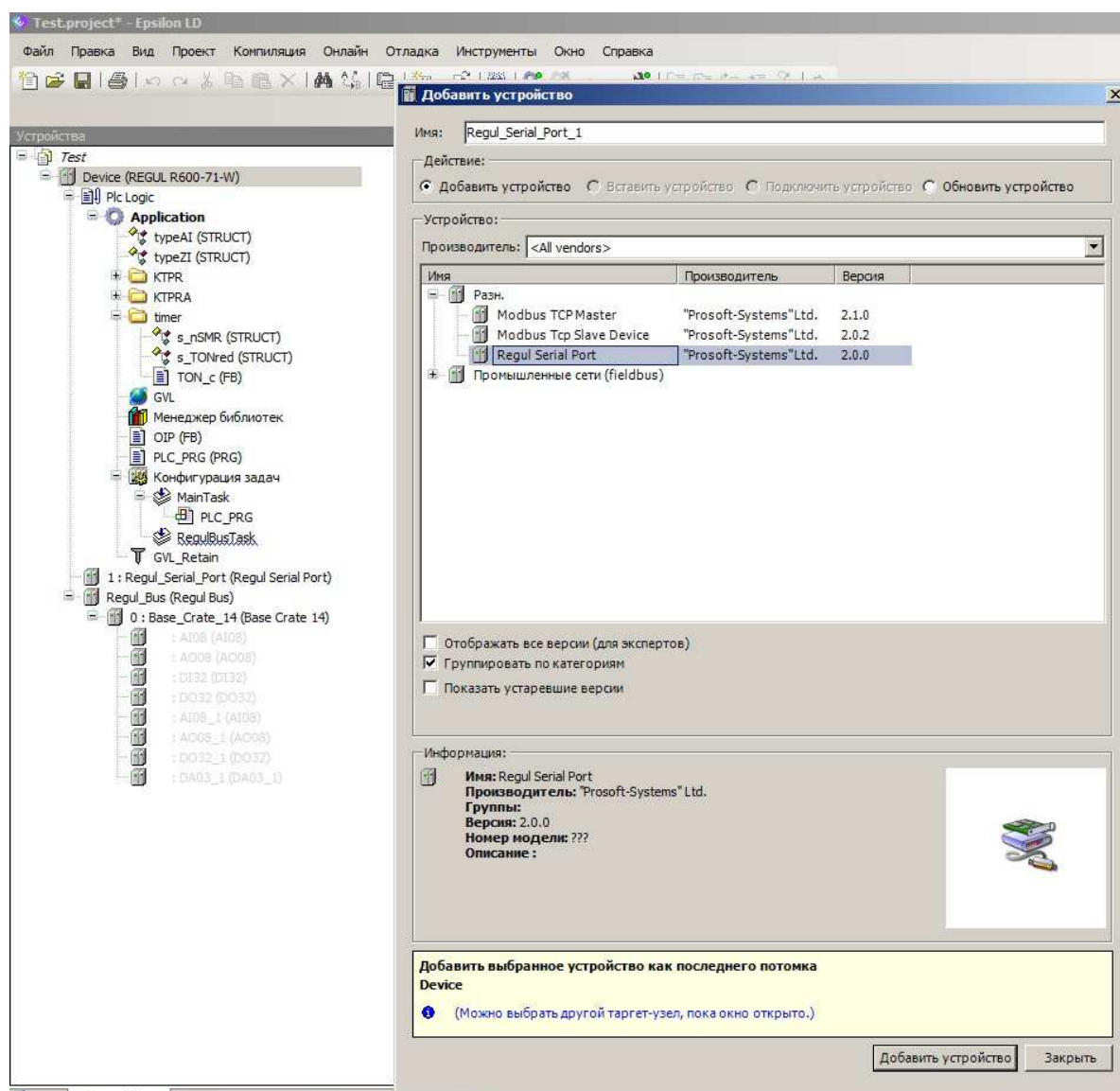
Настройки ntp

Настройка Modbus RTU Master

Шаг 1. Добавление последовательного порта

Настройка Modbus выполняется одинаково как для Regul R600, так и для Regul R500. Предполагается, что в проекте уже добавлен контроллер семейства Regul R600 или Regul R500. Далее следует решить, какие последовательные порты будем использовать для опроса. Первый вариант – какой-либо из портов RS-485 (RS-232) на процессорном модуле. В этом случае, в дереве устройств, выбираем устройство типа Regul R500 или Regul R600 и по клику правой кнопкой мыши выбираем «Добавить устройство», находим и добавляем Regul Serial

Port. Разрешается добавить не более двух устройств данного типа (по количеству физических портов RS-485 (RS-232) на процессорном модуле).



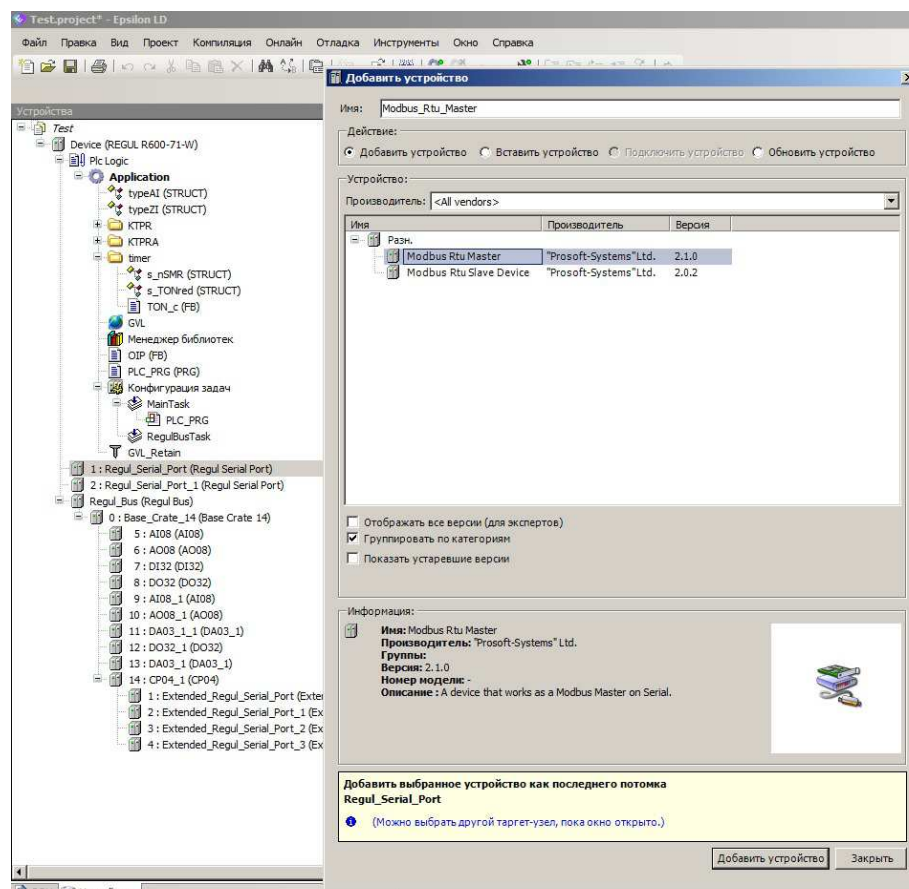
Добавление устройства «Regul serial port»

Если для опроса решено использовать порты отдельного модуля CP04, этот модуль должен быть предварительно добавлен в крейт. Далее для модуля добавляем дочернее устройство «Extended REGUL Serial Port». Поддерживается до четырех устройств данного типа на каждый модуль CP04. После добавления порта двойным щелчком на вновь добавленном устройстве открываем настройку, где можно указать скорость обмена (baudrate), четность (parity), количество бит данных (databits) и прочие параметры.

Шаг 2. Назначение Modbus RTU Master.

После описания необходимых последовательных портов, следует выбрать какой-либо из них и добавить в качестве дочернего устройства Modbus RTU Master. Действия

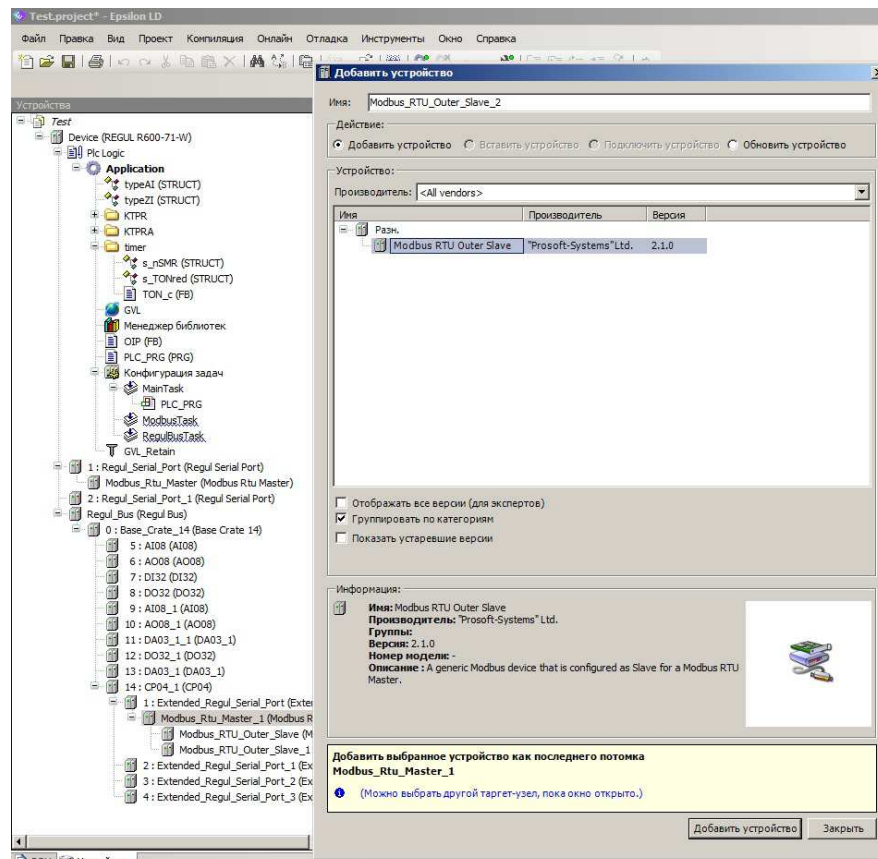
аналогичны предыдущим – щелчок правой кнопкой мыши на соответствующем узле в перечне устройств и вызов команды «Добавить устройство». Дополнительные настройки не требуются.



Добавление устройства Modbus RTU Master

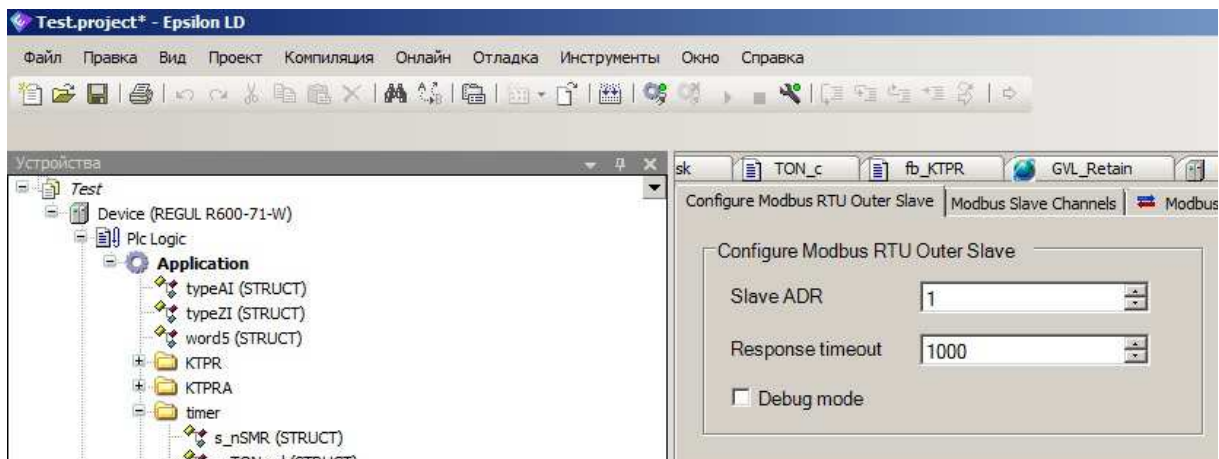
Шаг 3. Добавление slave-устройств.

После назначения Modbus RTU Master, также как в предыдущих пунктах, вызываем для него диалог «Добавить устройство», выбираем Modbus RTU Outer Slave и нажимаем кнопку «Добавить устройство». Не закрывая диалогов, можем добавить несколько slave-устройств. Максимальное количество ограничено диапазоном доступных адресов Modbus. Закрыв диалог, двойным щелчком на выбранном slave-устройстве открываем его настройку.



Добавление устройства Modbus RTU Outer Slave

Далее следует указать адрес Modbus и время ожидания ответа в мс.



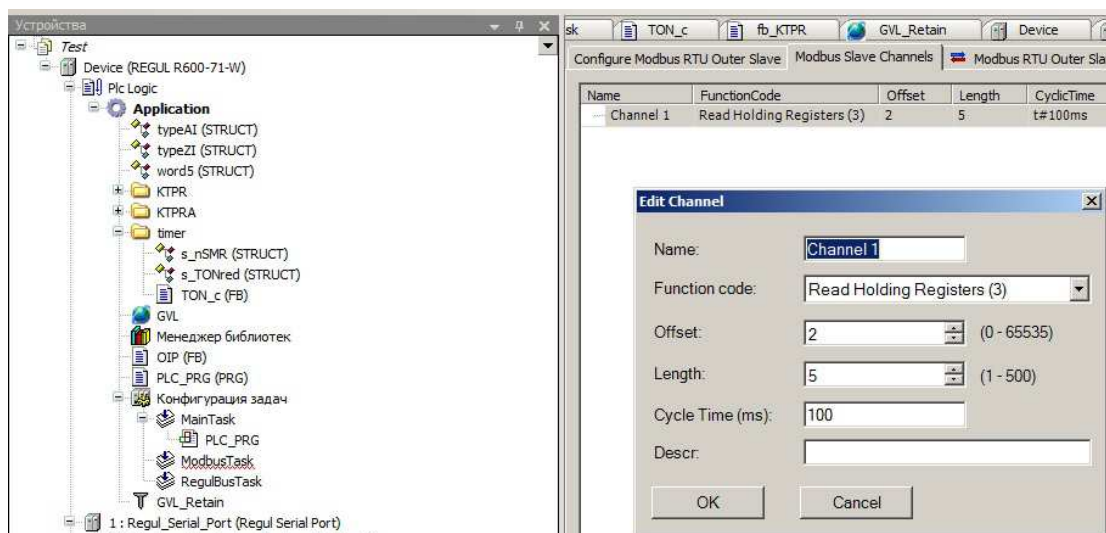
Редактор устройства Modbus RTU Outer Slave

Шаг 4. Настройка каналов – опрашиваемых регистров Modbus.

Каждый канал (channel) представляет собой описатель, задающий набор регистров Modbus (стартовый и количество), функцию для чтения/записи, периодичность опроса в мс., а также привязку к какой-либо переменной прикладной программы. Связанная переменная

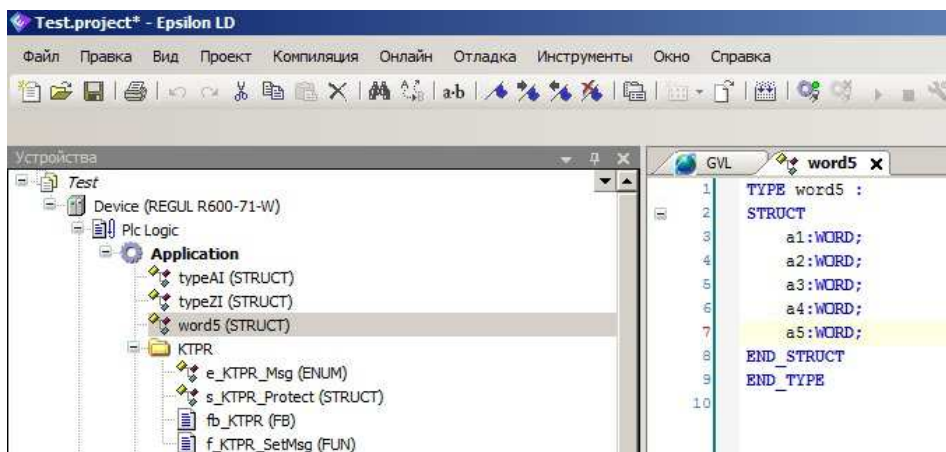
может быть как глобальной (определенной в GVL), так и указанной в отдельном программном блоке (POU).

Нажимаем кнопку «Add Channel» (Настройка выбранного slave, вкладка Modbus Slave Channels). На примере показан диалог «Edit channel», в котором мы задали канал для чтения функцией 3 Modbus 5 регистров, начиная со второго, с периодичностью 100 мс.

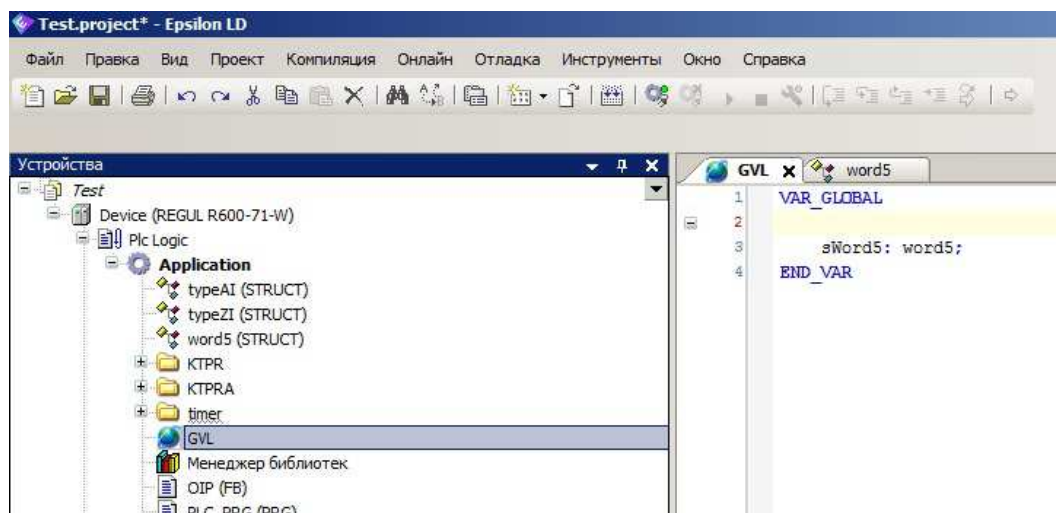


Добавление нового канала

Созданный канал можно связать с переменной любого атомарного типа (byte, word, int и т. д.), размер которой совпадает (не больше) размера читаемого блока данных Modbus. Более универсальным является подход, когда мы привязываем к каналу не атомарные переменные, а экземпляры структур, описанных в проекте. Размер структуры (его можно посмотреть функцией SIZEOF) также не должен превышать объема читаемых по каналу данных. Для примера создадим структуру Word5 из пяти двухбайтовых полей.



Затем создаем глобальную переменную sWord5 (экземпляр структуры Word5)



Далее в настройке «Modbus RTU Slave Соотнесение входов/выходов» связываем весь блок данных с переменной sWord5. Для этого, нажимая на кнопку , вызываем диалог «Ассистент ввода», в котором находим требуемую переменную.



Ассистент ввода

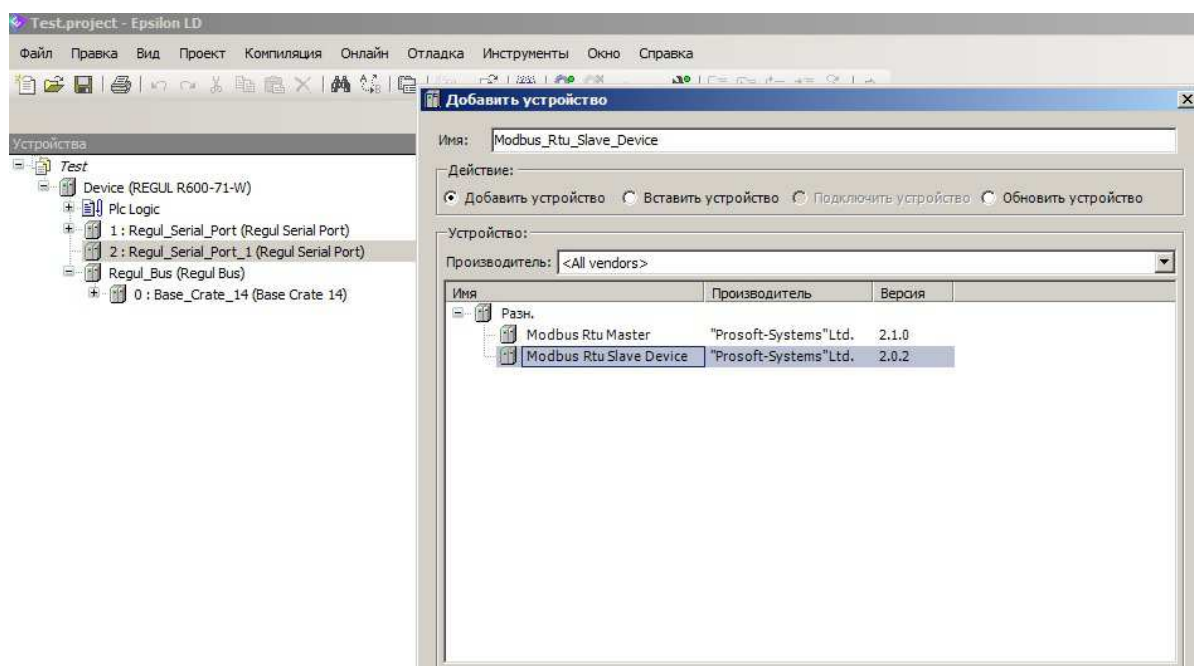
Далее по аналогии настраиваем необходимое количество каналов и их привязку к переменным.

Настройка Modbus RTU Slave

Шаг 1. Добавление Slave устройства

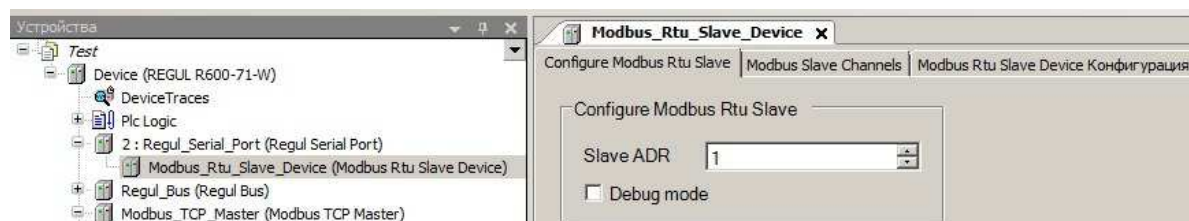
Предполагается, что в проекте уже добавлен контроллер семейства Regul R600 или Regul R500, а также добавлен последовательный порт для обмена. Эта процедура описана в разделе «Настройка Modbus RTU Master»

На выбранном устройстве «Regul Serial Port» или «Extended Regul Serial Port» по клику правой кнопкой мыши выбираем «Добавить устройство», находим и добавляем «Modbus Rtu Slave Device»



Добавление устройства «Modbus Rtu Slave Device»

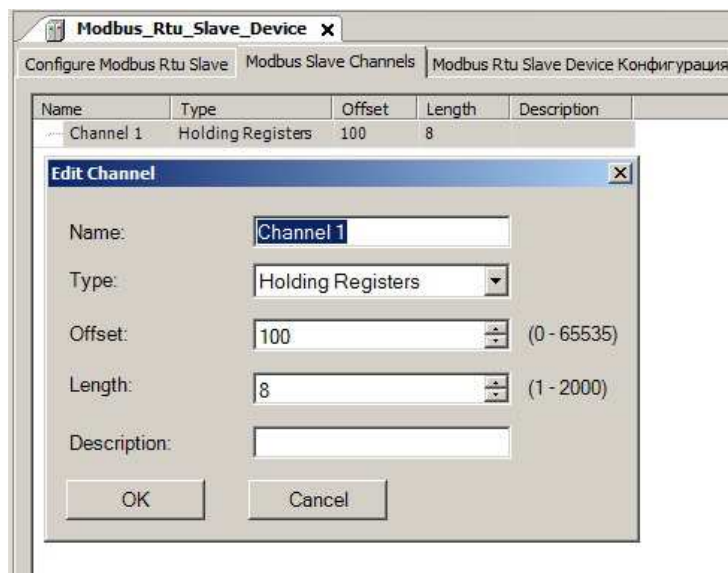
После того как «Modbus Rtu Slave Device» появился в списке устройств, двойным кликом мыши вызываем его редактор, где на вкладке «Configure Modbus Rtu Slave» назначаем ему Slave адрес.



Редактор устройства Modbus Rtu Slave Device

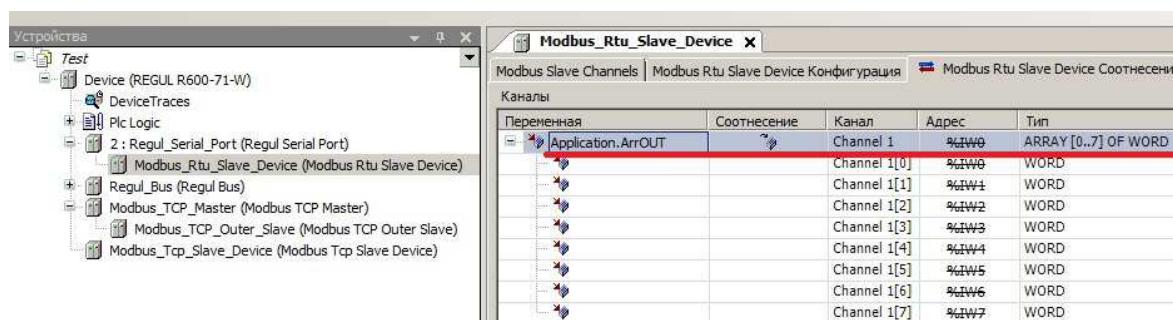
Шаг 2. Настройка каналов (Channels)

Далее описываем каналы (Channels) – группы регистров Modbus, которые можно прочитать/записать. На вкладке Modbus Slave Channels, нажимаем кнопку «Add Channel» для добавления, либо «Edit» для изменения канала. В появившемся окне диалога («Add Channel» или «Edit channel») указываем тип хранимых данных (Type), стартовый регистр (Offset) и количество хранимых элементов данных (Length).



Добавление нового канала

Если выбранный тип канала «Discrete Inputs» или «Coils», то поле Length показывает количество флагов (битовых значений). Если тип «Input Registers» или «Holding Registers», то поле Length показывает количество регистров – двухбайтовых блоков. Содержимое одного регистра может быть представлено переменной типа WORD в Epsilon LD.



Соотнесение переменных

После создания списка каналов следует связать их с переменными программы ПЛК. Для этого открываем настройку «Modbus Rtu Slave Соотнесение входов/выходов», двойной клик в соответствующей строке колонки «Переменная» и, нажимая на кнопку , вызываем диалог

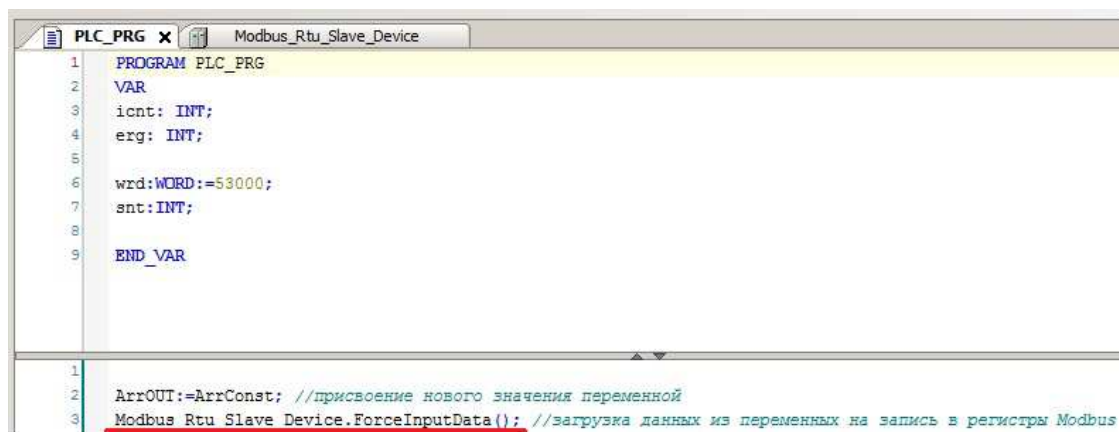
«Ассистент ввода», в котором находим требуемую переменную. На рисунке показано, что канал из 8 регистров Modbus сопоставлен с переменной ArrOut, которая является массивом из 8 элементов типа WORD (двухбайтовое целое).

Шаг 3. Дополнительные настройки

Согласно спецификации Modbus данные контроллера, которые определяются как «Discrete Inputs» и «Input Registers», предназначены только для чтения. В этом случае при связывании какой-либо переменной программы с регистрами канала Modbus Slave, она будет считаться переменной «для чтения», то есть после каждой итерации цикла ПЛК, ее содержимое копируется в область данных Modbus и может быть получено подключившимся Modbus-клиентом.

Если мы определили тип канала как «Coils» или «Holding Registers», то такие данные согласно спецификации Modbus допускают и чтение и запись. Но в Epsilon LD, согласно стандарту программирования ПЛК IEC 61131-3, переменные, связанные с внешними устройствами, могут быть либо «только для чтения», либо «только для записи». В последнем случае данные, прочитанные с внешних устройств, перед каждой итерацией программы ПЛК, копируются в связанные с ними переменные.

Текущая версия Epsilon LD связывает данные канала «Coils» или «Holding Registers» с переменными в режиме «только для записи». Для того чтобы реализовать чтение этих данных, необходимо в прикладной программе на ПЛК, после присвоения переменной нового значения, вызвать для данного Slave-устройства метод «ForceInputData».



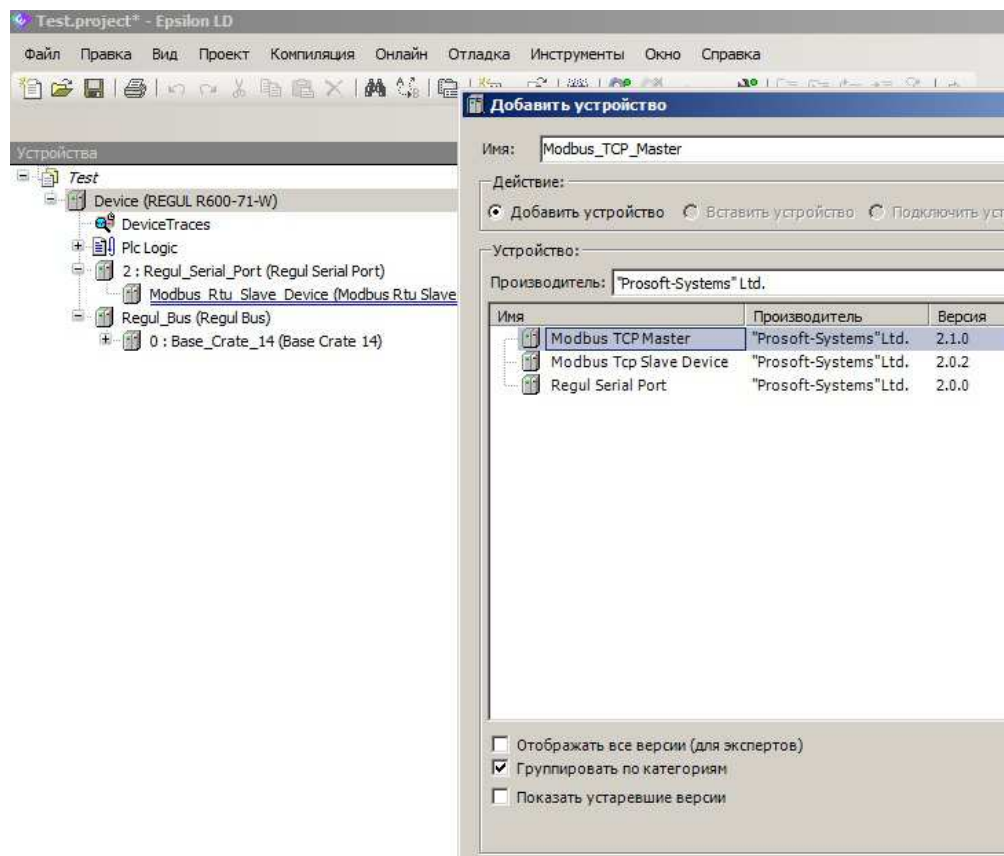
Вызов метода «ForceInputData»

В этом случае значение переменной будет принудительно скопировано в регистры Modbus и в следующей итерации отослано master-устройству.

Настройка Modbus TCP Master

Шаг 1. Назначение Modbus TCP Master

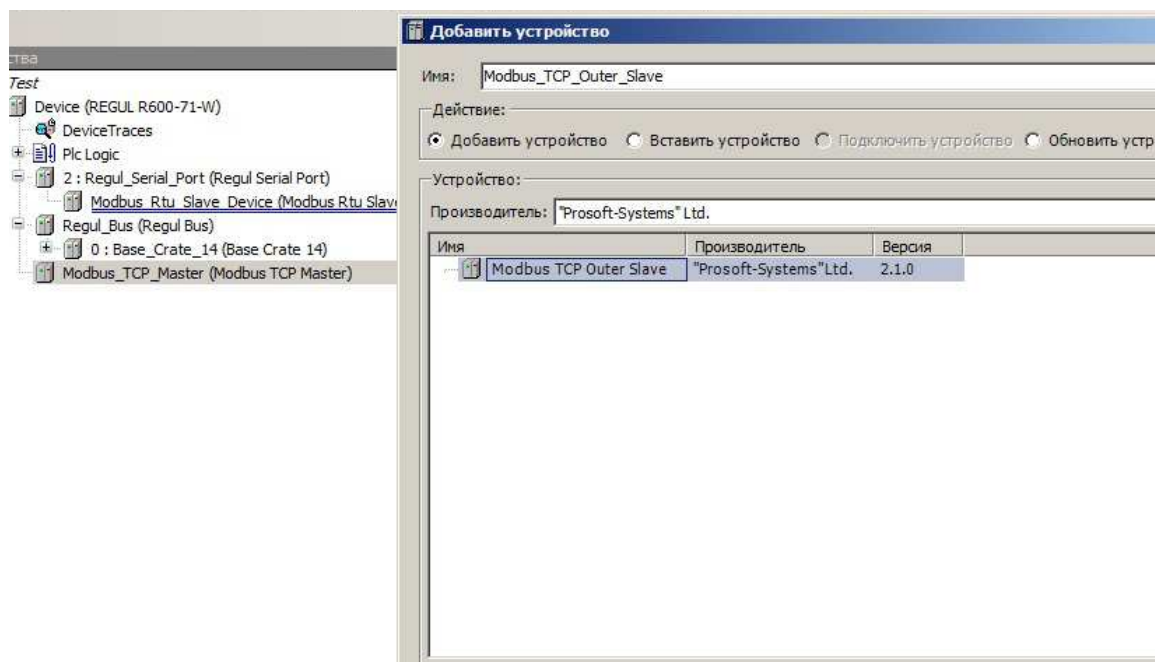
Предполагается, что в проекте уже описан контроллер семейства REGUL R600 или REGUL R500. Выбираем устройство типа Regul R500 или Regul R600 и, по клику правой кнопкой мыши, выбираем «Добавить устройство», находим и добавляем «Modbus TCP Master».



Добавление устройства «Modbus TCP Master»

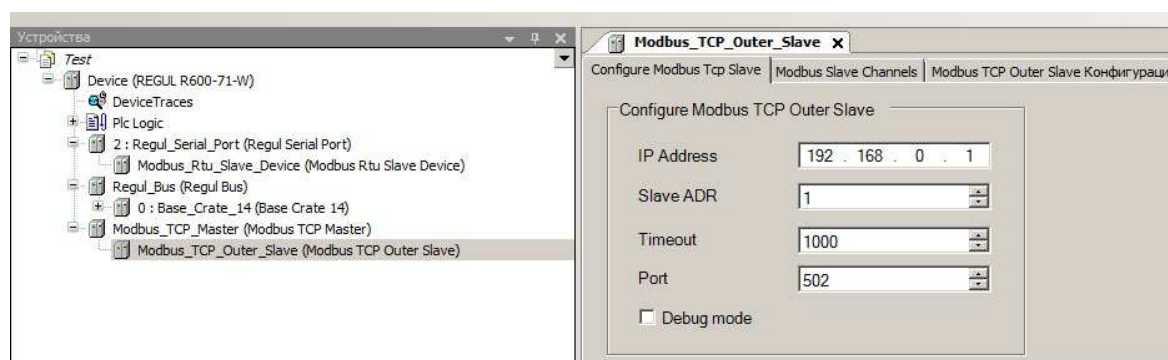
Шаг 2. Добавление внешних slave-устройств

После назначения Modbus TCP Master к нему, в качестве дочернего устройства, следует добавить Modbus TCP Outer Slave (также через вызов команды «Добавить устройство» на соответствующем объекте). Можно добавить несколько устройств данного типа.



Добавление устройства «Modbus TCP Outer Slave»

Для каждого внешнего slave-устройства следует задать четыре параметра: ip-адрес, Modbus Slave адрес, таймаут ожидания ответа и номер порта (сокета TCP).



Редактор устройства «Modbus Tcp outer slave»

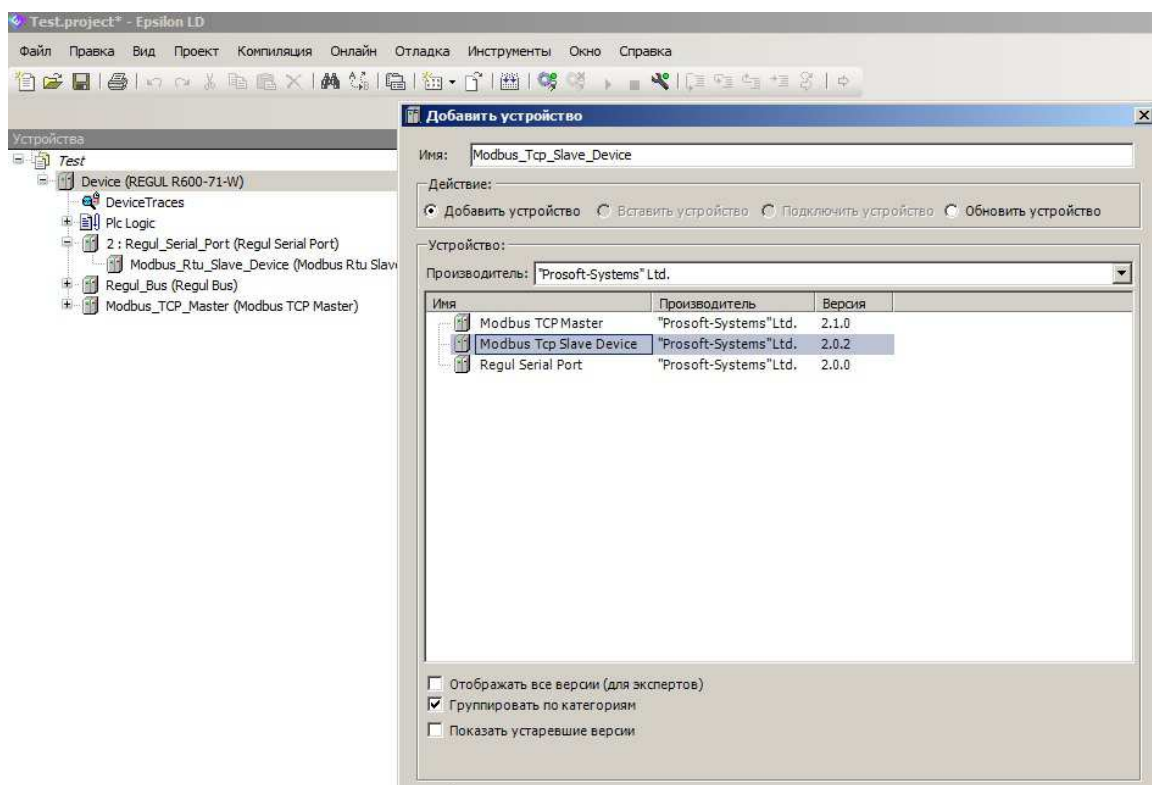
Шаг 3. Настройка каналов – опрашиваемых регистров Modbus.

Каждый канал (channel) представляет собой описатель, задающий набор регистров Modbus (стартовый и количество), функцию для чтения/записи, периодичность опроса в мс., а также привязку к какой-либо переменной прикладной программы. Связанная переменная может быть как глобальной (определенной в GVL), так и указанной в отдельном программном блоке (POU). Процедура настройки каналов аналогична, как и для Modbus RTU Master, которая описана в соответствующем разделе.

Настройка Modbus TCP Slave

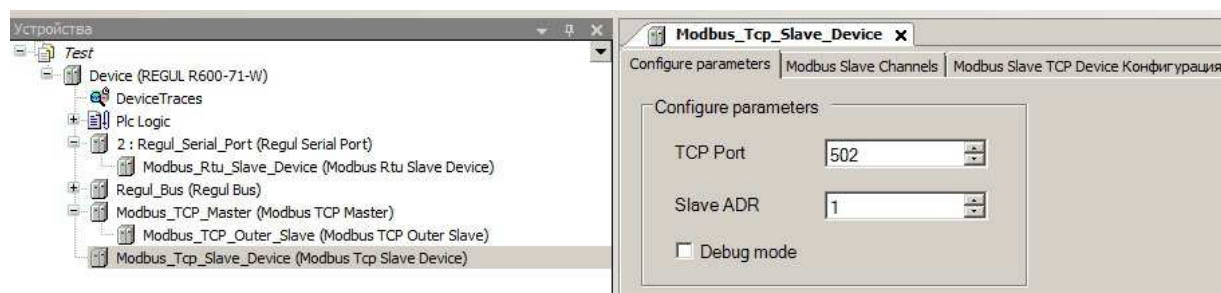
Шаг 1. Добавление Slave устройства

Предполагается, что в проекте уже описан контроллер семейства REGUL R600 или REGUL R500. В окне просмотра «Устройства» выбираем устройство типа Regul R500 или Regul R600 и, по клику правой кнопкой мыши, выбираем «Добавить устройство», находим и добавляем «Modbus TCP Slave Device».



Добавление устройства «Modbus Tcp Slave Device»

После того как «Modbus Tcp Slave Device» появился в списке устройств, двойным кликом мыши вызываем меню его настройки, где на вкладке «Configure parameters» назначаем ему Slave адрес и входящий TCP порт, который будет прослушиваться для приема запросов от master-клиентов.



Редактор устройства «Modbus Tcp Slave Device»

Шаг 2. Настройка каналов – опрашиваемых регистров Modbus.

Далее на вкладке «Modbus Slave Channels» потребуется создать список каналов (опрашиваемых регистров Modbus) и связать его с переменными программы ПЛК. Эти настройки аналогичны тем, которые описаны в разделе «Настройка Modbus RTU Slave»

ПРИЛОЖЕНИЕ А.

ПАРАМЕТРЫ МОДУЛЕЙ

Все параметры модулей контроллеров REGUL R600 разбиты на группы. Каждая группа соответствует определенному аспекту настройки модуля или его «физического» канала.

Если модуль имеет один или более каналов ввода/вывода или коммуникационных каналов, то параметры модуля для каждого из этих каналов также объединены в отдельные группы (далее под буквой «N» в названии таких групп подразумевается номер канала, к которому эта группа относится).

Системные параметры (Service Data)

Группа системных параметров встречается во всех типах модулей REGUL R600. Эти параметры имеют атрибут *«Только для чтения»* и используются средой исполнения контроллера для идентификации самого модуля и его типа. Использование этих параметров в прикладной программе не рекомендуется.

Общие параметры канала (Channel N generic settings)

Группа общих параметров канала есть почти у всех каналов ввода/вывода или коммуникационных каналов в модулях REGUL R600. Эта группа служит для задания значения маскирования «физического» канала модуля.

Таблица А.1 – Список общих параметров канала

Параметр	Название в среде разработки	Тип данных	Значение по умолчанию	Описание
Маскирование	Is Channel Masked	Bool	False	Задаёт маскирование канала. False – не замаскирован, True – замаскирован.

Параметры канала аналогового ввода

Все модули аналогового ввода REGUL R600 имеют богатый встроенный функционал обработки входных сигналов. К ним относятся:

- 2 стадии преобразования измеряемой величины: из кода АЦП в значение электрической величины и из электрической величины в значение инженерной величины;

- усреднение измеряемого значения (с конфигурируемым временем усреднения);
- бракование канала при слишком резком «скачке» инженерной величины;
- бракование канала по выходу электрической или инженерной величины за заданные пределы.

Таблица А.2 – Список параметров канала аналогового ввода

Параметр	Название в среде разработки	Тип данных	Значение по умолчанию	Описание
Коэффициент K_0	K0 coefficient	Float	0.0	Коэффициент K_0 преобразования электрической величины в инженерную величину.
Коэффициент K_1	K1 coefficient	Float	1.0	Коэффициент K_1 преобразования электрической величины в инженерную величину.
Коэффициент k_0 (+/- 10В)	k0 coefficient (+/- 10 V)	Float	ПЗУ	Коэффициент k_0 преобразования кода АЦП в напряжение для канала +/- 10 В.
Коэффициент k_1 (+/- 10В)	k1 coefficient (+/- 10 V)	Float	ПЗУ	Коэффициент k_1 преобразования кода АЦП в напряжение для канала +/- 10 В.
Коэффициент k_0 (0 - 10 В)	k0 coefficient (0 - 10 V)	Float	ПЗУ	Коэффициент k_0 преобразования кода АЦП в напряжение для канала 0 - 10 В.
Коэффициент k_1 (0 - 10 В)	K1 coefficient (0 - 10 V)	Float	ПЗУ	Коэффициент k_1 преобразования кода АЦП в напряжение для канала 0 - 10 В.
Коэффициент k_0 (4 - 20 мА)	k0 coefficient (4 - 20 mA)	Float	ПЗУ	Коэффициент k_0 преобразования кода АЦП в ток в диапазоне 4-20 мА.
Коэффициент k_1 (4 - 20 мА)	k1 coefficient (4 - 20 mA)	Float	ПЗУ	Коэффициент k_1 преобразования кода АЦП в ток в диапазоне 4-20 мА.
Тип канала	Channel type	Byte	0	Тип канала: 0 – +/- 10 В; 1 – от 0 до 10 В; 2 – от 4 до 20 мА.
Мгновенное «электрическое значение»	Instant electrical value	Float		Откалиброванное физическое значение (величина электрического сигнала). Используется при калибровке канала.
Коэффициент усреднения α	Averaging time in milliseconds	UInt	0	Коэффициент усреднения α в диапазоне значений [0...1]
Скорость скачка	Maximum rate of value change	Float	3.4E+38	Максимальная скорость изменения инженерной величины за 660 мкс (время скачка).

Время нечувствительности к скачку	Channel discarding coefficient	Byte	100	Время нечувствительности к скачку, мс
Нижняя граница инженерной величины	Lower boundary value	Float	-3.4E+38	Значение нижней границы инженерной величины
Верхняя граница инженерной величины	Upper boundary value	Float	3.4E+38	Значение верхней границы инженерной величины.

Преобразование входной величины

Значение кода АЦП, полученное после оцифровки входного значения канала, проходит две стадии преобразования: из кода АЦП в электрическую величину и из электрической величины в инженерную. Под электрической величиной здесь понимается значение электрического сигнала, измеряемого каналом (т.е. величина тока или напряжения). Под значением инженерной величины понимается значение измеряемого каналом параметра технологического процесса (температура, давление, скорость течения и т.д.).

Преобразование кода АЦП в электрическую величину производится по следующей формуле:

$$U = k_0 + k_1 C,$$

где U – электрическая величина, C – код АЦП, k_0 и k_1 – коэффициенты преобразования.

Коэффициенты k_0 и k_1 – по сути, являются калибровочными коэффициентами канала. Их значение вычисляется и записывается в ПЗУ модуля на стадии его производства. В последующем, в процессе эксплуатации модуля, при необходимости перекалибровать канал, значение этих коэффициентов могут быть заново вычислены и перезаписаны.

Функция преобразования «электрической» величины в инженерную позволяет, во-первых, разгрузить от этой логики прикладную программу, а во-вторых, использовать встроенный функционал модуля по бракованию инженерной величины. Преобразование «электрической» величины в инженерную производится по следующей формуле:

$$X = K_0 + K_1 U,$$

где X – значение инженерной величины, U – полученная электрическая величина, K_0 и K_1 – коэффициенты преобразования.

Коэффициенты K_0 и K_1 зависят от используемого для измерений датчика и задаются пользователем при конфигурировании модуля.

Если использование встроенного функционала модуля по бракованию не требуется, или если датчик обладает нелинейной характеристикой, то значения этих коэффициентов необходимо оставить неизменными. В этом случае, в качестве измеренных значений, модуль будет передавать в прикладную программу значение электрической величины.

Усреднение измеряемого значения

Усреднение измеряемого значения производится с использованием функции экспоненциально взвешенного скользящего среднего.

Для конфигурирования алгоритма усреднения используется параметр «Время усреднения». Этот параметр задает время в миллисекундах, за которое усредненное значение достигнет 95 % от новой величины сигнала, при его одномоментном изменении, при условии, что значение сигнала и усредненной величины до этого были равны.

Бракование входных сигналов

Возможность автоматического бракования сигналов в модулях аналогового ввода REGUL R600 применяется, когда необходимо автоматически обнаруживать аномальные значения или скорости изменения измеряемых сигналов. При браковании сигнала, модуль автоматически устанавливает общий флаг бракования канала и флаг соответствующий критерию, по которому сигнал был отбракован.

Поддерживаются два типа бракования: «по скачку» и по выходу за границу.

Бракование «по скачку»

Возможность бракования «по скачку» используется для бракования сигнала при слишком резком изменении инженерной величины.

Для конфигурирования алгоритма бракования сигнала «по скачку» используются следующие параметры:

- «Максимальная скорость изменения инженерной величины». Значение этого параметра задает максимальную возможную разницу инженерной величины между двумя идущими подряд отсчетами (660 мкс.).
- «Коэффициент бракования». Значение этого параметра задает время (в процентах от времени усреднения) от начала «скачка», через которое канал будет забракован.

Бракование по выходу за границы

Бракование сигнала по выходу за границы производится по трем видам границ: интервал измерений АЦП, интервал возможного изменения «электрического» сигнала и интервал возможного изменения инженерной величины.

Интервал измерений АЦП модуля зависит от установленного АЦП и не может быть сконфигурирован пользователем.

Интервал возможного изменения «электрического» сигнала зависит от типа измеряемого данным каналом электрического сигнала. Например, если канал работает в режиме измерения тока 4 – 20 мА, границы этого интервала будут равны, соответственно 4 и 20 мА. При понижении значения тока ниже 4 мА или выше 20 мА канал будет забракован. Значения границ этого интервала также не могут быть сконфигурированы пользователем.

Границы интервал возможного изменения инженерной величины задаются пользователем и определяются спецификой технологического процесса.

Параметры канала аналогового вывода

Модули аналогового вывода REGUL R600 так же, как и модули аналогового ввода, поддерживают автоматическое преобразование инженерной величины в электрическую величину и далее в значение в кодах АЦП. Коэффициенты этих преобразований задаются в группе параметров канала аналогового вывода.

Таблица А.3 – Список параметров канала аналогового вывода

Параметр	Название в среде разработки	Тип данных	Значение по умолчанию	Описание
Коэффициент K_0	K0 coefficient	Float	0.0	Коэффициент K_0 преобразования инженерной величины в электрическую.
Коэффициент K_1	K1 coefficient	Float	1.0	Коэффициент K_1 преобразования инженерной величины в электрическую.
Коэффициент k_0	k0 coefficient	Float	ПЗУ	Коэффициент k_0 преобразования электрической величины в код ЦАП.
Коэффициент k_1	k1 coefficient	Float	ПЗУ	Коэффициент k_1 преобразования электрической величины в код ЦАП.
Таймаут управления каналом при потере связи с мастером		UInt	0	Таймаут управления каналом при потере связи с модулем ЦП, мс. Диапазон [1 – 65535] (0 – бесконечность)

Предустановленное значение канала при потере связи		Float	0.0	Предустановленное значение канала при потере связи с модулем ЦП
Состояние канала при потере связи с мастером		Byte	0	Состояние канала при потере связи с модулем ЦП: 0 – установить предустановленное значение 1 – не изменять состояние

Параметры канала дискретного ввода

Модули дискретного ввода REGUL R600 поддерживают защиту от «дребезга» и инверсию входного сигнала.

Таблица А.4 – Список параметров канала дискретного ввода

Параметр	Название в среде разработки	Тип данных	Значение по умолчанию	Описание
Время «антидребезга»	Rebouncing protection time in milliseconds	UInt	0	Значение времени «антидребезга» в миллисекундах.
Инверсия канала	Inversion	Bool	False	Инверсия канала.
Передавать мастеру информацию о событиях	ChangeStateEvent Enable	Bool	False	Даёт возможность из прикладной программы получать метки времени при смене значения канала

Порядок работы с метками времени от модуля DI32

- 1) Выставить опцию генерировать события у модуля DI32.
- 2) Обработать массив событий из кода прикладной программы следующим образом:

```

FOR ch := 0 TO 31 DO
  IF DI_32_011.Events[ch].Count > 0 THEN
    FOR e := 0 TO DI_32_011.Events[ch].Count - 1 DO
      DI_32_011.Events[ch].E[e].xValue;
      DI_32_011.Events[ch].E[e].ulITS;
      //printf("DI32 channel=%d change value to %d on %s", ch,
      DI_32_011.Events[ch].E[e].xValue,
      UTCTIMENS_TO_STRING(DI_32_011.Events[ch].E[e].ulITS, TRUE, TRUE));
    END_FOR
    DI_32_011.Events[ch].Clear();
  END_IF
END_FOR;
```

Параметры канала дискретного вывода

Таблица А.5 – Список параметров канала дискретного вывода

Параметр	Название в среде разработки	Тип данных	Значение по умолчанию	Описание
Таймаут потери связи с ЦП	Security state timeout in tens of milliseconds	UInt	0	Таймаут управления каналом при потере связи с мастером, мс. Диапазон [1 – 65535] (0 – бесконечность)
Состояние канала при потере связи с мастером	Security strategy	Byte	0	Состояние канала при потере связи с мастером: 0 – не изменять состояние; 1 – установить 0; 2 – установить 1.

Параметры канала последовательного порта

Таблица А.6 – Список параметров канала последовательного порта

Параметр	Название в среде разработки	Тип данных	Значение по умолчанию	Описание
Скорость	Bandwidth	UInt	9600	Скорость обмена.
Биты данных	Data bits count	Byte	8	Количество бит данных.
Проверка четности	Check parity	Byte	0	Проверка четности: 0 – нет проверки (none), 1 – устанавливается при нечетности (even), 2 – устанавливается при четности (odd), 3 – всегда 1 (mark), 4 – всегда 0 (space).
Стоповые биты	Stop bits count	Byte	0	Количество стоповых бит: 0 – 1, 1 – 1.5, 2 – 2.

ПРИЛОЖЕНИЕ Б

ВХОДЫ/ВЫХОДЫ МОДУЛЕЙ

Каждый модуль ввода/вывода контроллеров REGUL R600 имеет определенное количество логических каналов ввода/вывода, к которым можно привязать переменные прикладной программы. Некоторые из этих логических входов/выходов соответствуют тем или иным «физическим» входам/выходам модуля, а некоторые привязаны к внутренним регистрам модуля.

Как и в случае с параметрами модулей, входы/выходы также можно разбить на группы. Под буквой «N» в названии таких групп подразумевается номер канала, к которому эта группа относится

Системные входы/выходы

Почти все модули REGUL R600 обладают одним или двумя системными входами/выходами: так называемые heartbeat-счетчики. Эти входы/выходы используются системным ПО контроллеров для определения наличия связи между модулем и центральным процессором. Использовать эти входы/выходы в прикладной программе не рекомендуется.

Таблица Б.1 – Список системных входов/выходов

Название входа/выхода	Название в среде разработки	Тип данных	Направление передачи	Описание
Heartbeat-счетчик ЦП	Master heartbeat	Byte	Выход	Системный счетчик, значение которого генерируется ЦП. Используется для определения наличия связи с ЦП в модуле.
Heartbeat-счетчик модуля	Module heartbeat	Byte	Вход	Системный счетчик, значение которого генерируется модулем. Используется для определения наличия связи с модулем в ЦП.

Входы/выходы модулей аналогового ввода

Каждому каналу в модулях аналогового ввода REGUL R600 сопоставляется один вход, соответствующий рассчитанному значению инженерной величины, входной флаг, соответствующий статусу бракования канала, и шесть входных флагов, соответствующих причинам бракования.

Таблица Б.2 – Список входов/выходов канала аналогового ввода

Название входа/выхода	Название в среде разработки	Тип данных	Направление передачи	Описание
Входное значение	Input N	Real	Вход	Рассчитанное модулем входное значение инженерной величины.
Бракование канала	Discarded	Bit	Вход	Общий флаг бракования канала.
Выход сигнала за верхнюю границу измерения инженерной величины	Upper technical bound exceeded	Bit	Вход	Канал забракован по выходу сигнала за верхнюю границу измерения инженерной величины
Выход сигнала за нижнюю границу измерения инженерной величины;	Lower technical bound exceeded	Bit	Вход	Канал забракован по выходу сигнала за нижнюю границу измерения инженерной величины
Выход сигнала за верхнюю границу измерения электрической величины	Upper electrical bound exceeded	Bit	Вход	Канал забракован по выходу сигнала за верхнюю границу измерения электрической величины
Выход сигнала за нижнюю границу измерения электрической величины	Lower electrical bound exceeded	Bit	Вход	Канал забракован по выходу сигнала за нижнюю границу измерения электрической величины
Недостоверность канала по выходу за верхнюю границу АЦП	Upper ADC validity bound exceeded	Bit	Вход	Канал забракован по недостоверности канала по выходу за верхнюю границу АЦП
Недостоверность канала по выходу за нижнюю границу АЦП	Lower ADC validity bound exceeded	Bit	Вход	Канал забракован по недостоверности канала по выходу за нижнюю границу АЦП

Входы/выходы модулей аналогового вывода

Модули аналогового вывода REGUL R600 автоматически преобразуют подаваемое на выход значение из инженерной величины в величину электрического сигнала, если при конфигурировании модуля были заданы соответствующие коэффициенты.

Таблица Б.3 – Список входов/выходов канала аналогового вывода

Название входа/выхода	Название в среде разработки	Тип данных	Направление передачи	Описание
Выходное значение	Output N	Real	Выход	Выходное значение инженерной величины.

Все модули аналогового вывода используют в своей работе внешний источник питания. Если на клеммах модуля, предназначенных для подключения источника питания, нет напряжения, модуль устанавливает флаг отсутствия внешнего питания.

Таблица Б.4 – Список диагностических входов/выходов канала аналогового вывода

Название входа/выхода	Название в среде разработки	Тип данных	Направление передачи	Описание
Отсутствует внешнее питание	No outer power supply	Bit	Вход	Нет напряжения на клеммах источника внешнего питания.

Входы/выходы модулей дискретного ввода

Таблица Б.5 – Список входов/выходов канала дискретного ввода

Название входа/выхода	Название в среде разработки	Тип данных	Направление передачи	Описание
Входное значение	Input N	Bit	Вход	Входное значение канала.

Входы/выходы модулей дискретного вывода

Таблица Б.6 – Список входов/выходов канала дискретного вывода

Название входа/выхода	Название в среде разработки	Тип данных	Направление передачи	Описание
Выходное значение	Output N	Bit	Выход	Выходное значение канала.