ООО КБ «АГАВА»
620026, г. Екатеринбург,
ул. Бажова, 174, 3-й эта.
т/ф 343/262-92-76 (78,87)
http://www.kb-agava.ru

РУКОВОДСТВО ПРОГРАММИСТА CODESYS

АГСФ.421445.005 РП

Редакция 1.8

Екатеринбург

2022

Оглавление

1. Введение

Руководство по эксплуатации содержит сведения, необходимые для обеспечения правильной эксплуатации и полного использования технических возможностей программируемых логических контроллеров АГАВА ПЛК-40, АГАВА ПЛК-50, АГАВА ПЛК-60 далее по тексту ПРИБОР, ПЛК или КОНТРОЛЛЕР.

Разработка приложений для контроллеров серии АГАВА ПЛК предполагает использование среды разработки Codesys 3.5.

1.1. Используемые термины и сокращения

SDK – Software development kit – комплект средств разработки приложений;
SSH – Secure Shell – протокол защищенного подключения;
ВМ – виртуальная машина
ОЗУ – оперативное запоминающее устройство;
ОС – операционная система;
ПО – программное обеспечение;
ПЛК – программируемый логический контроллер (промышленный контроллер);
ПКМ – правая кнопка мыши;
СП - среда программирования;
ФБ – функциональный блок;
ФНЧ – фильтр нижних частот;
ФС – файловая система.

1.2. Полезные ссылки

- Сайт КБ АГАВА – www.kb-agava.ru
- Описание программируемого логического контроллера АГАВА ПЛК-40 – http://www.kb-agava.ru/kontrollery/kontrollery_universalnogo_primeneniya/agava-plc-40
- Описание программируемого логического контроллера АГАВА ПЛК-50 – http://www.kb-agava.ru/kontrollery/kontrollery_universalnogo_primeneniya/agava-plc-50
- Описание программируемого логического контроллера АГАВА ПЛК-60 – http://www.kb-agava.ru/kontrollery/kontrollery_universalnogo_primeneniya/agava-plc-60
- Руководство по эксплуатации АГАВА ПЛК-40 – http://www.kb-agava.ru/index.php?route=module/product_downloads/get&did=340
- SDK АГАВА ПЛК-30, ПЛК-40, ПЛК-50, ПЛК-60 среда программирования Codesys – http://files.kb-agava.ru/index.php/s/F9Uuk5PLiMYP6Cv
- Конфигуратор аппаратных средств АГАВА ПЛК-40 – http://www.kb-agava.ru/agava-plc-40-zakaz

- Конфигуратор аппаратных средств АГАВА ПЛК-60 – http://www.kb-agava.ru/agava-plc-60-zakaz
- Руководство по эксплуатации АГАВА МВВ-40 – http://www.kb-agava.ru/index.php?route=module/product_downloads/get&did=341
- Утилита настройки параметров АГАВА МВВ-40 – http://www.kb-agava.ru/index.php?route=module/product_downloads/get&did=342
- Презентация контроллеров серии «АГАВА ПЛК-40» – https://www.youtube.com/watch?v=DX3asi1PK1M
- Описание работы со средой разработки Codesys 3.x на русском языке находится в файле – C:\Program Files (x86)\3S CODESYS\CODESYS\Online Help\ru\codesys.chm
  Этот файл становится доступен после установки Codesys 3.5.

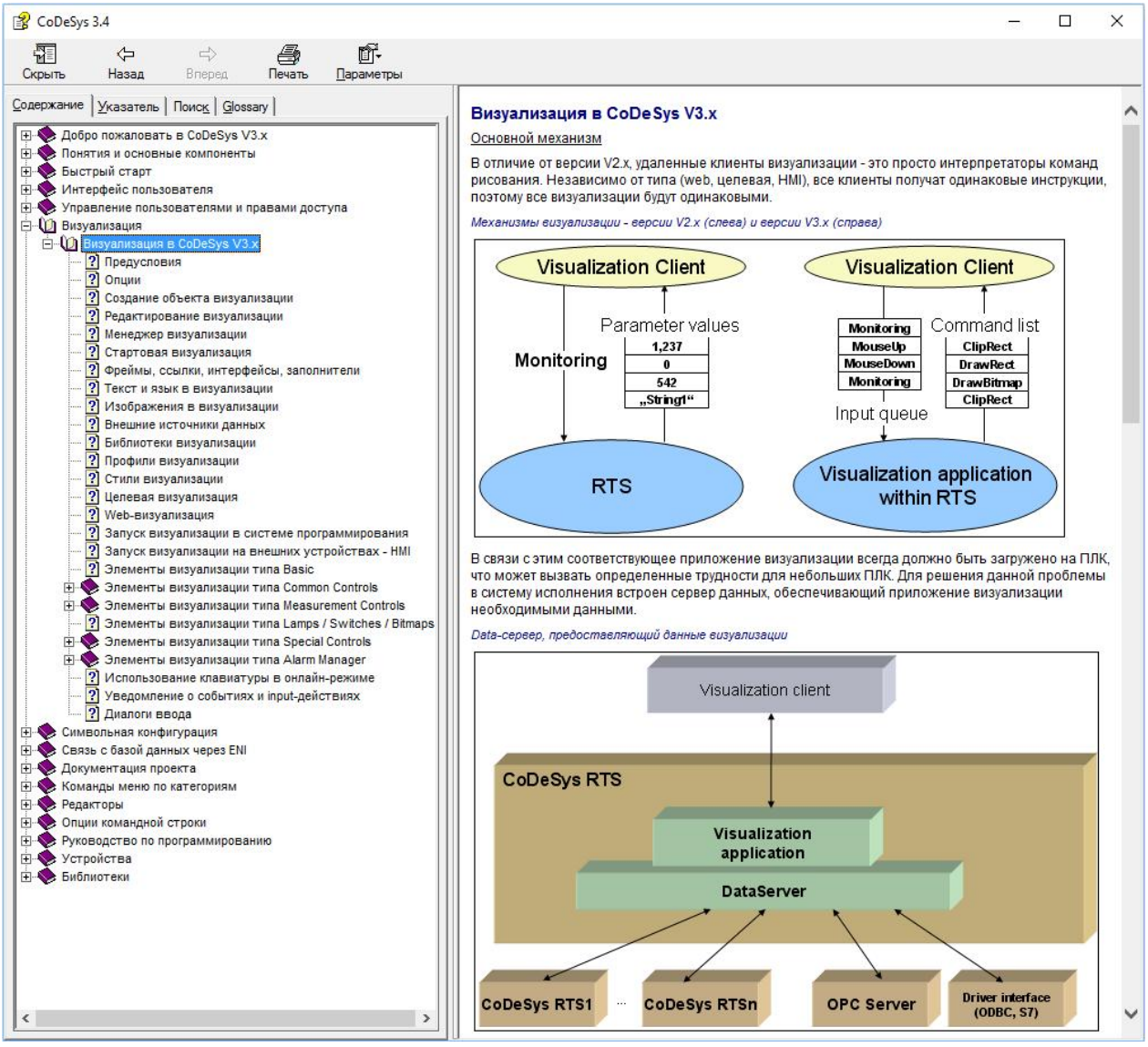


Рисунок 1-1. Справка по Codesys 3.x

2. Назначение

Программируемый логический контроллеры серии АГАВА ПЛК предназначены для создания систем автоматизированного управления технологическим оборудованием в различных областях промышленности, жилищно-коммунального и сельского хозяйства.

Логика работы ПЛК определяется потребителем в процессе программирования контроллера.

2.1. Использование комплекта средств разработки (SDK)

Программирование контроллера осуществляется с помощью среды разработки проекта CODESYS v3.5 SP14.

При использовании среды программирования Codesys 3.5 SP10:

- в свойствах проектах установить компилятор 3.5.**10**.0.

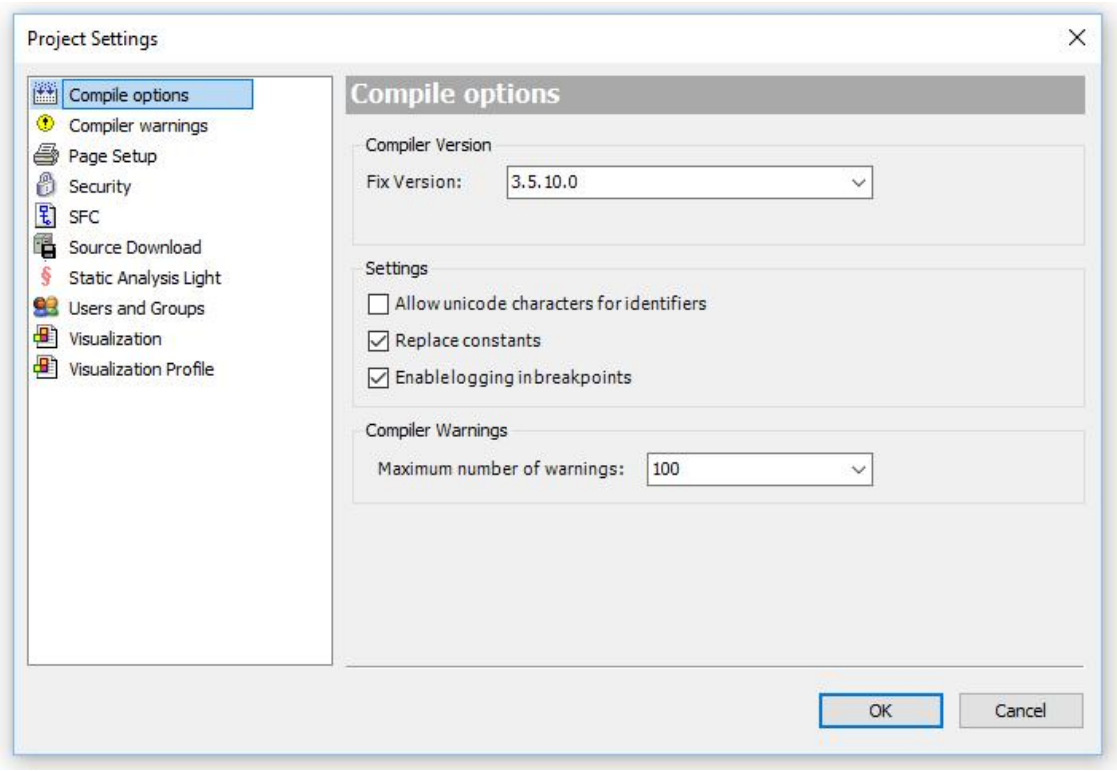


Рисунок 2-1. Установка версии компилятора

- в свойствах проекта установить профиль визуализации 3.5.10.0;

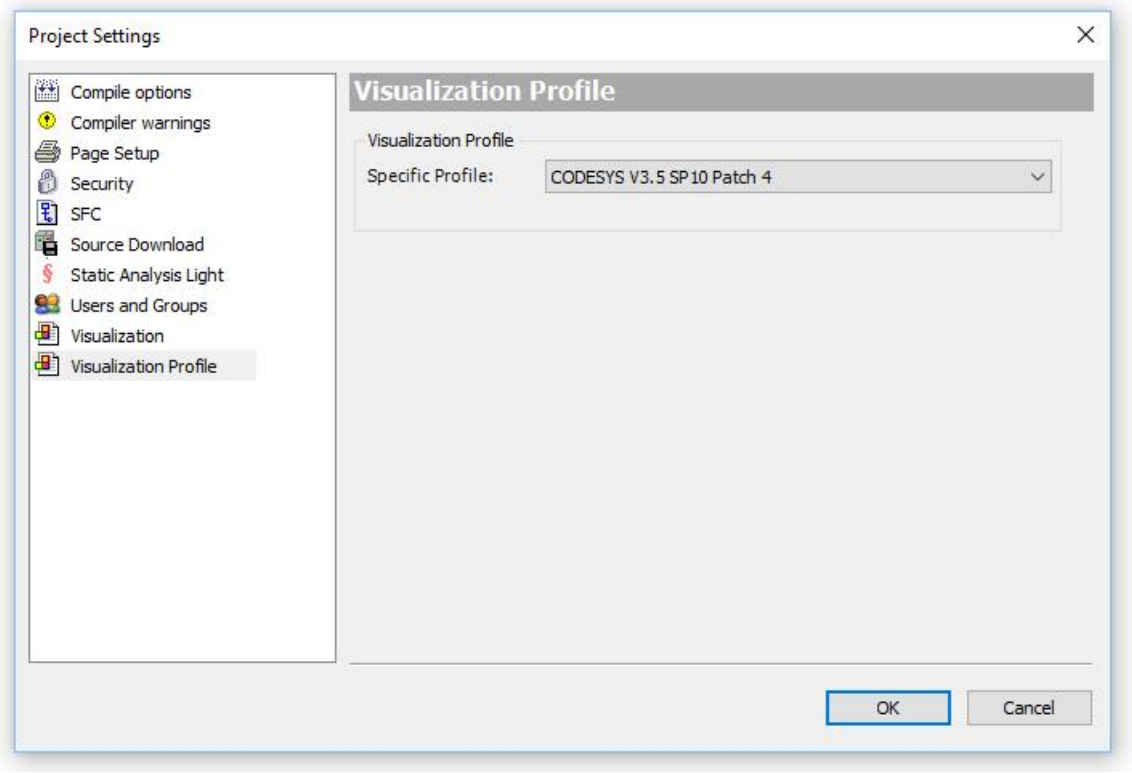


Рисунок 2-2. Установка профиля визуализации

При использовании Codesys 3.5 SP14:

- в свойствах проектах установить компилятор 3.5.**14.**10.

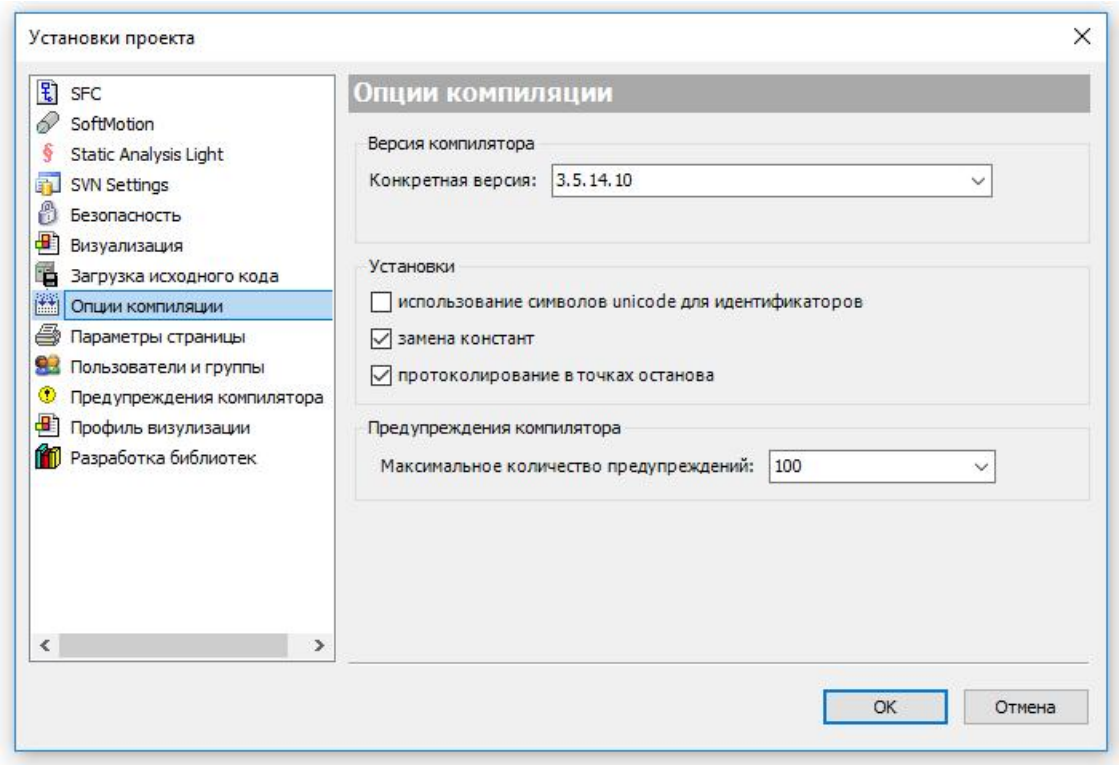


Рисунок 2-3. Установка версии компилятора

- в свойствах проекта установить профиль визуализации 3.5.14.10;

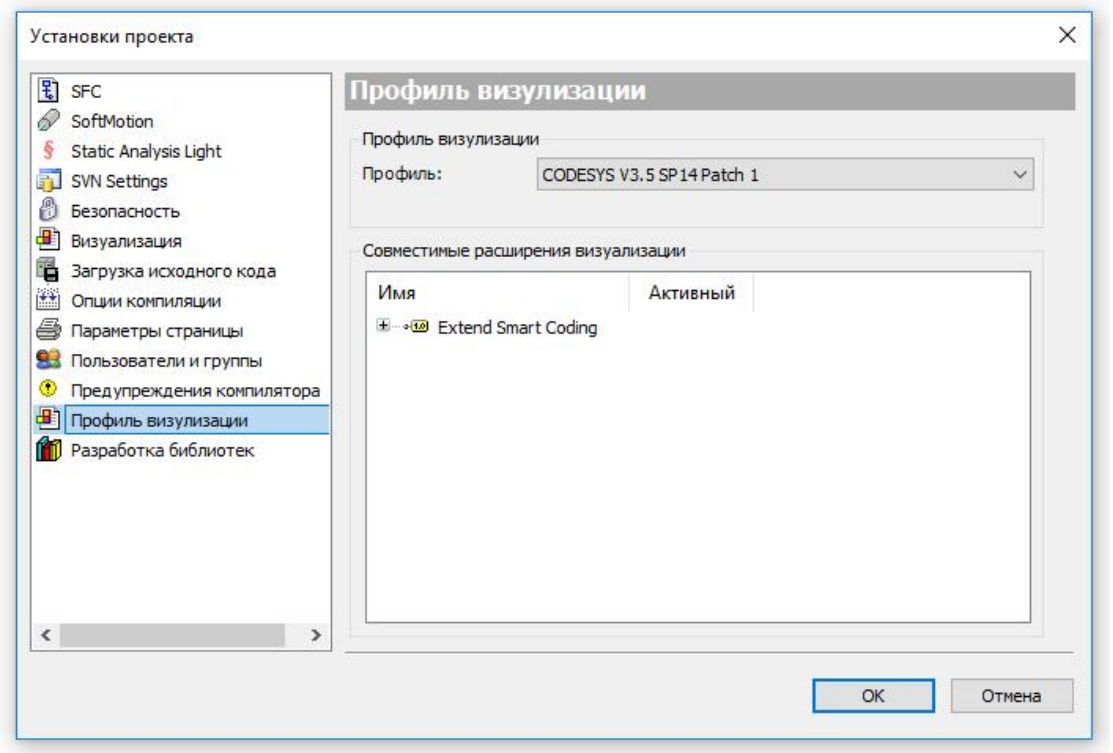


Рисунок 2-4. Установка профиля визуализации

Проекты могут быть разработаны с использованием любого из языков стандарта IEC 61131-3: SFC (Sequential Function Chart), FBD (Function Block Diagram), LD (Ladder Diagram), IL (Instruction List), ST (Structured Text), а также языка CFC (Continuous Function Chart).

Практически все примеры, входящие в состав Agava SDK, созданы и сохранены в Codesys 3.5.10.40 (SP10 Patch 4) и Codesys 3.5.14.10 (SP14 Patch 1). Эти версии является рекомендуемыми для использования. Новые версии потенциально также являются рабочими, но не тестировались на совместимость с Agava SDK.

Порядок работы с примерами из Agava SDK:

1) открыть проект;
2) Выбрать файл описания устройства;
3) подключиться к ПЛК;
4) удалить загруженный в ПЛК проект;
5) выполнить команду очистки проекта;
6) выполнить компиляцию проекта;
7) загрузить проект в ПЛК;
8) запустить проект.

Удаление проекта необходимо выполнять при помощи команды «Сброс заводской устройства». Вызывается из контекстного меню при нажатии ПКМ на устройстве в дереве проектов. В результате действия этой команды проект удаляется полностью.

Так как примеры, входящие в состав Agava SDK создавались в разных версиях Codesys 3.5.10.40 (SP10 Patch 4) и Codesys 3.5.14.10 (SP14 Patch 1), в некоторых случаях возможны появления ошибок при компиляции, самая распространённая ошибка - различие версии компиляторов и версии профиля визуализации, возникает при открытии в Codesys 3.5 SP14 проекта созданного в более ранних версиях Codesys 3.5, например Codesys 3.5 SP10.

Рисунок 2-5. Различие версии компилятора и профиля визуализации

Для устранения данной ошибки , необходимо в свойствах проекта открытого в версии Codesys 3.5 SP14 установить версию компилятора 3.5.14.10, установить версию профиля визуализации CODESYS V3.5 SP14 Patch 1, далее выполнить команду «компиляция\очистить все», затем команду «компиляция генерировать код».

2.2. Выбор файла описания устройства

В SDK включены файлы описания всех устройств производства КБ АГАВА, использующих СП CODESYS.

Для разработки приложений необходимо выбрать файл описания, соответствующий используемому устройству.

В SDK для CODESYS 3.5 SP10 включены файлы описания:

- «Agava» – АГАВА ПЛК-30 и ПЛК-60 без web-визуализации.
- «Agava WV» – АГАВА ПЛК-30 и ПЛК-60 с web-визуализацией.
- «Agava TV» – АГАВА ПЛК-40 и ПЛК-50 без web-визуализации.
- «Agava TV+WV» – АГАВА ПЛК-40 и ПЛК-50 с web-визуализацией.

3. Предустановленный проект

Контроллер поставляется с тестовым проектом Codesys, который показывает расположение и состав модулей расширения. Для каждого модуля на отдельном экране можно посмотреть: номер слота, в который модуль установлен, статистику обмена и значения входов/выходов. Также имеется возможность управлять выходными сигналами.

Рисунок 3-1. Вид экрана при запуске проекта на ПЛК

Ниже приведены информационные экраны для конфигурации, представленной на рисунке 3-1.

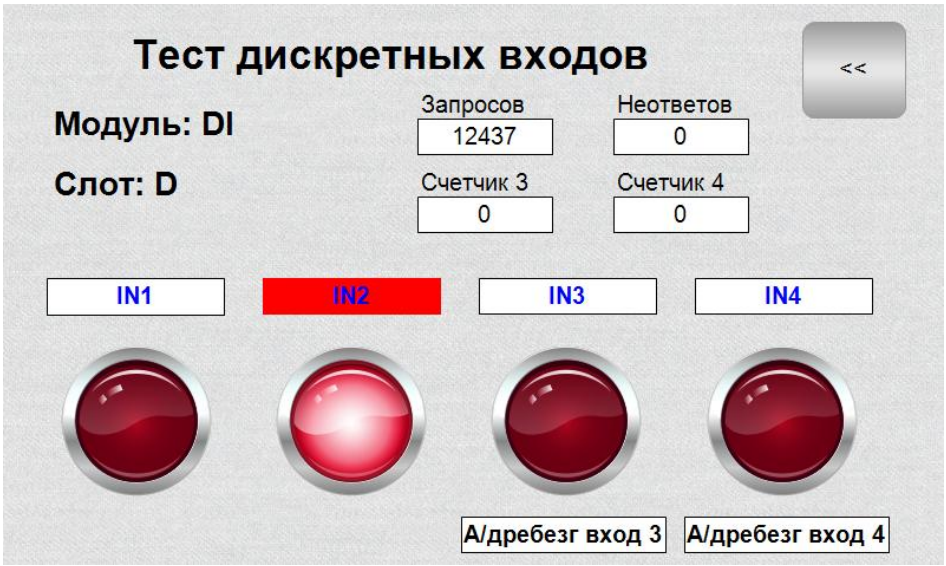


Рисунок 3-2. Экран тестирования модуля дискретных входов

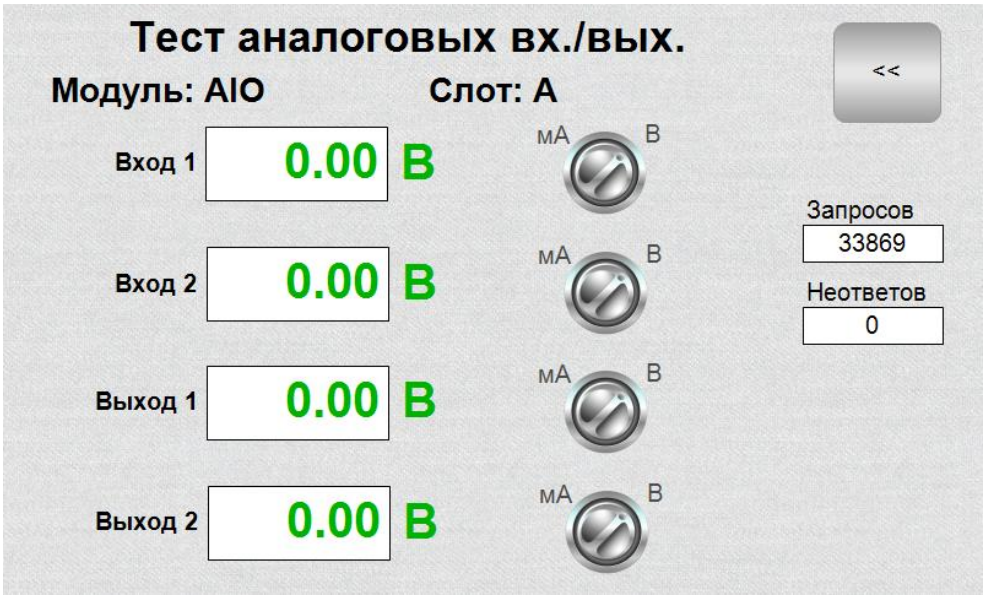


Рисунок 3-3. Экран тестирования модуля аналоговых входов/выходов

Рисунок 3-4. Экран тестирования модуля дискретных выходов типа «общий коллектор»

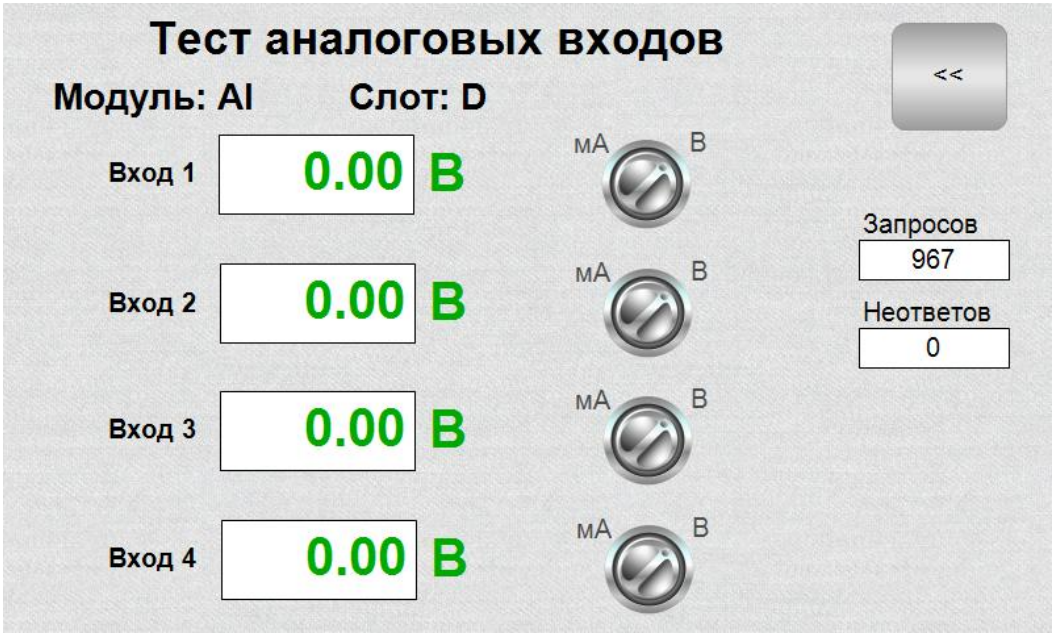


Рисунок 3-5. Экран тестирования модуля аналоговых входов

Рисунок 3-6. Экран тестирования модуля дискретных входов типа «симистор»

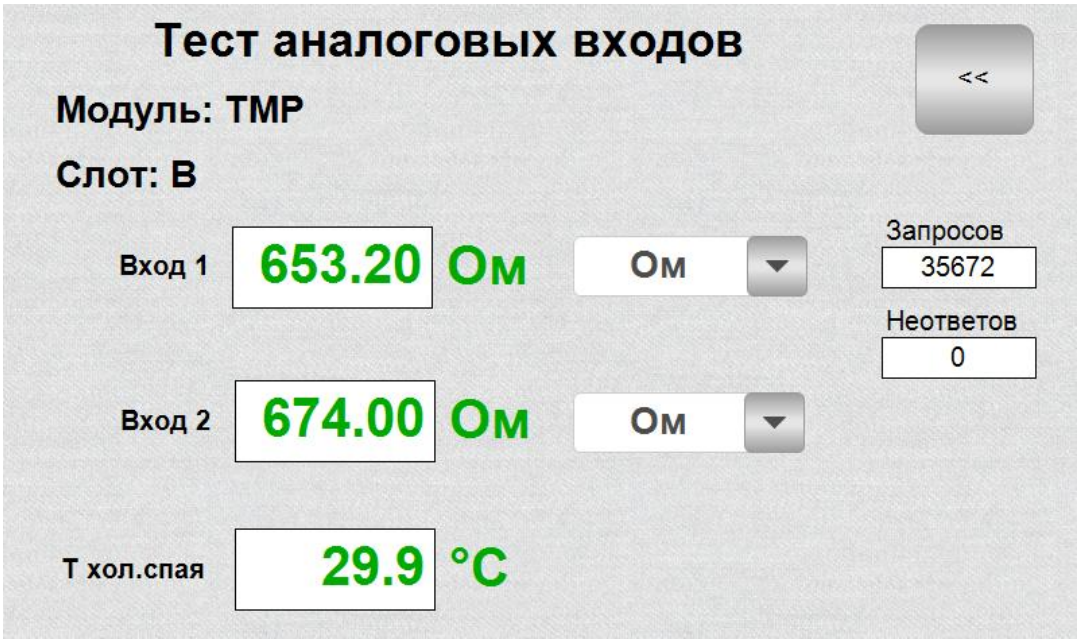


Рисунок 3-7. Экран тестирования модуля измерения температуры

Рисунок 3-8. Экран тестирования модуля дискретных входов типа «реле»

4. Разработка приложений в среде CODESYS V3.5

Детальное описание работы в среде программирования CODESYS приводится в документации, поставляемой вместе с СП. Ниже описывается основной порядок работы ПЛК с CODESYS.

4.1. Рекомендации по эксплуатации и разработке программного обеспечения

Обратите внимание!

При разработке прикладных программ для обеспечения максимального срока эксплуатации ПЛК рекомендуем контролировать частоту записи на внутреннюю flash-память ПЛК.

В ПЛК в качестве накопителя для хранения файлов операционной системы и пользовательских данных установлена flash-память, имеющая ограниченное количество циклов перезаписи.

Избегайте реализации алгоритмов **циклической** записи на внутреннюю память - "retain" переменных или записи данных через файловые операции. Ресурс памяти в конце концов будет исчерпан и ПЛК выйдет из строя.

В случаях, когда избежать наличия таких алгоритмов в разрабатываемом программном обеспечении невозможно, в качестве хранилища циклически изменяемых данных используйте SD-карту как можно большего объема и контролируйте целостность создаваемых на ней файлов. После выхода из строя SD-карты ее можно заменить на новую.

Для справки: в ПЛК-30 и ПЛК-60 установлена память типа SLC NAND, имеющая ресурс около 10000 циклов перезаписи одной ячейки. В ПЛК-40 и ПЛК-50 установлена память типа eMMC, также имеющая ресурс около 10000 циклов перезаписи одной ячейки, но за счет наличия специального контроллера, распределяющего записываемые на карту данные по всему свободному пространству, итоговый ресурс памяти выше в несколько раз чем у типа NAND, но тем не менее и он не бесконечен.

4.2. Начало работы

Для начала работы с ПЛК АГАВА необходимо:

1) Загрузить из хранилища комплект средств разработчика Agava SDK (см. раздел 1.2).
2) Установить пакет «AgavaLibraries.x.x.x.x.package» (x.x.x.x – номер версии), входящий в Agava SDK. Для этого нужно открыть менеджер пакетов в меню Tools, нажать кнопку Install и выбрать файл пакета в диалоговом окне. Пакет содержит в себе набор библиотек, драйверов и описаний устройств.
3) Установить драйвер RNDIS из Agava SDK с использованием файла «AgavaPLC. USB driver setup.exe». Этот драйвер, поддержка которого встроена в ОС Windows, создаёт виртуальную сетевую карту, доступную в Диспетчере устройств. На базе этой сетевой карты Windows создаёт сетевое подключение, работающее через USB подключение.

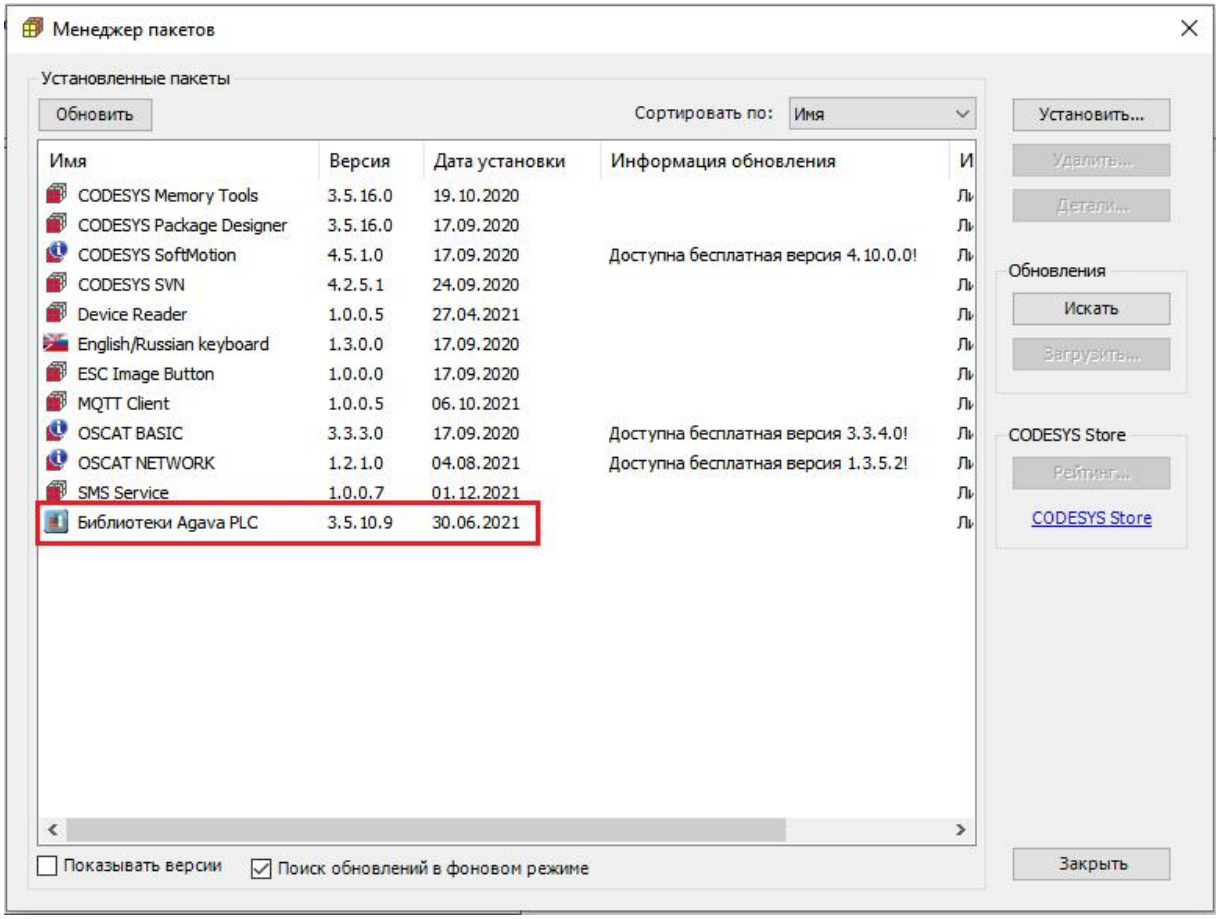


Рисунок 4- 1. Вид менеджера пакетов после установки

4.3. Создание нового проекта

Для создания нового проекта необходимо в СП CODESYS вызвать команду меню *File | New Project*, задать путь и имя проекта и выбрать подходящий шаблон Standard project.

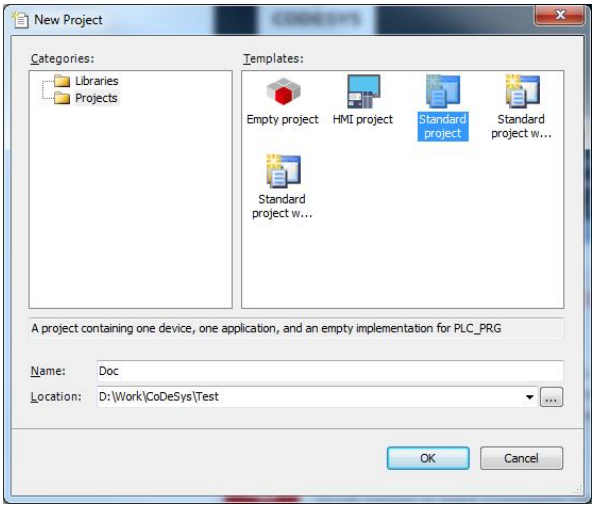


Рисунок 4- 2. Создание проекта

В списке устройств выбрать Ваш тип ПЛК, например: Agava TV, язык реализации программы ST.

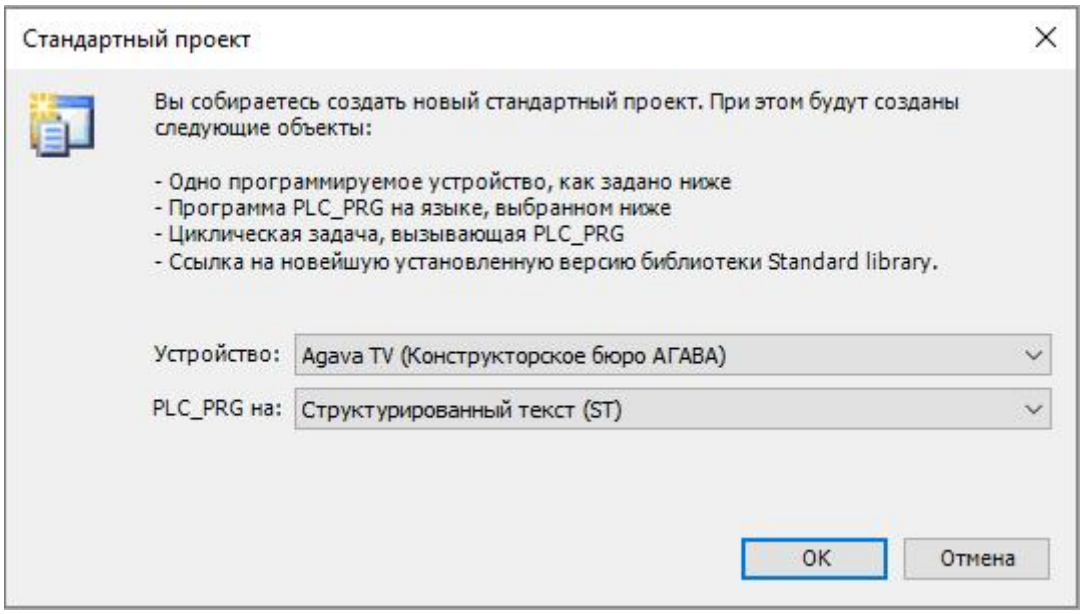


Рисунок 4- 3. Выбор описания ПЛК

4.4. Загрузка проекта в ПЛК

Для загрузки проекта в ПЛК необходимо:

1. Подключить ПЛК к компьютеру, на котором установлена СП CODESYS, через сеть Ethernet или с помощью USB кабеля из комплекта поставки.
2. Включить ПЛК, дождаться его загрузки.

Внимание!

Порты miniUSB и RS-232 не имеют гальванической развязки. Во избежание повреждения прибора, все подключаемое к нему оборудование (компьютер, сетевое оборудование, датчики и др.), имеющее клеммы заземления, должно быть надежно заземлено.

В окне Devices дважды щелкнуть на Device (Agava PLC-40). Откроется вкладка Device, в которой выбрать Scan Network для сканирования сети с подключенными ПЛК. После сканирования откроется список обнаруженных ПЛК.

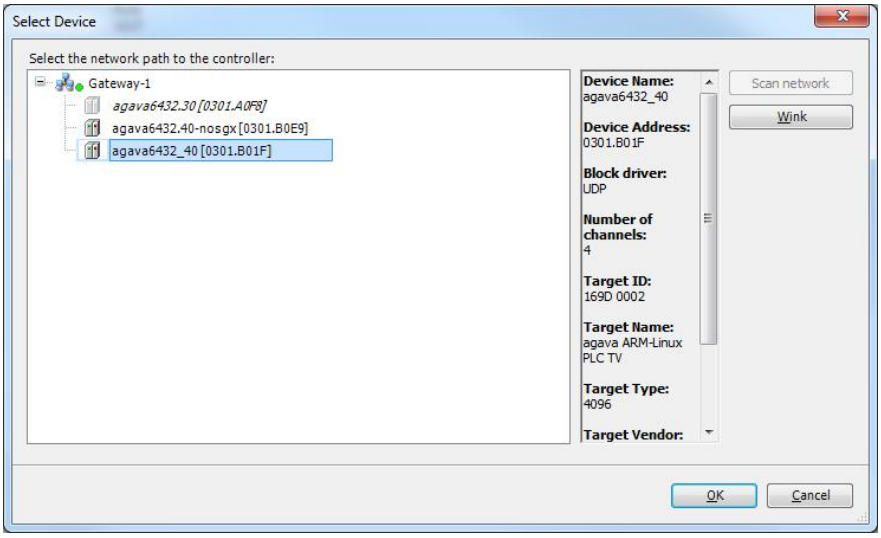


Рисунок 4- 4. Выбор ПЛК

Из списка выбрать ПЛК, к которому нужно подключиться, и нажать ОК. Далее выбрать *Online | Login* для подключения к ПЛК и загрузки проекта.

Рисунок 4- 5. Подключение СП CODESYS к ПЛК

Вы также можете указать IP-адрес ПЛК напрямую. Если подключение осуществляется с использованием USB кабеля, то для ПЛК IP-адрес имеет значение 192.168.7.1.

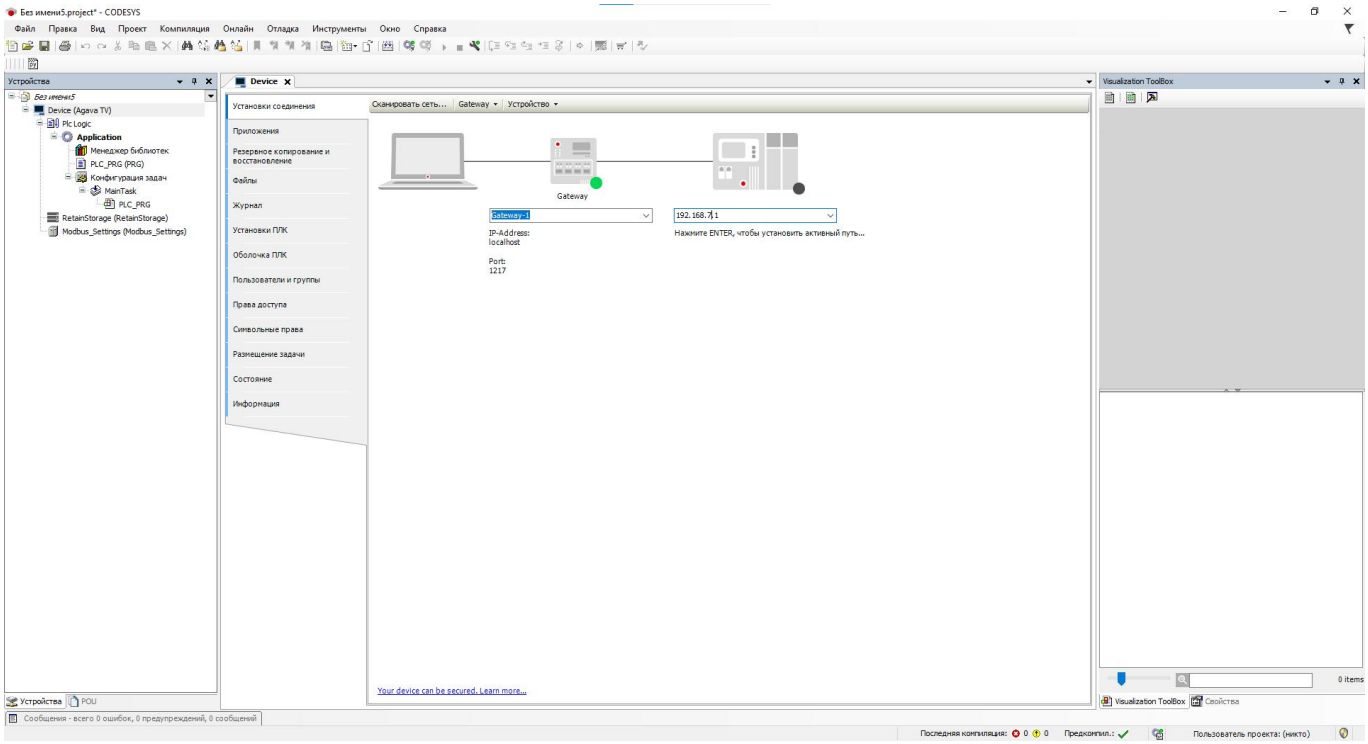


Рисунок 4- 6. Подключение СП CODESYS к ПЛК

5. Быстрый старт

5.1. Проект с использованием конфигуратора субмодулей ПЛК-40

Рассмотрим пример создания проекта с использованием ПЛК-40, содержащего субмодули дискретных входов, релейных выходов, аналоговых входов и входов термосопротивлений. В качестве примера конфигурации возьмём субмодули **DI**, **R**, **AI**, **TMP**. Проект рассматриваемого примера можно взять в SDK: Проекты\ПЛК-40\Быстрый старт\Быстрый_старт_конфигуратор.project

Добавим в проект корзину субмодулей ПЛК-40, для этого кликнем правой клавишей мыши по устройству **Device(Agava TV)** и в открывшемся контекстном меню выберем пункт «**Добавить устройство**».

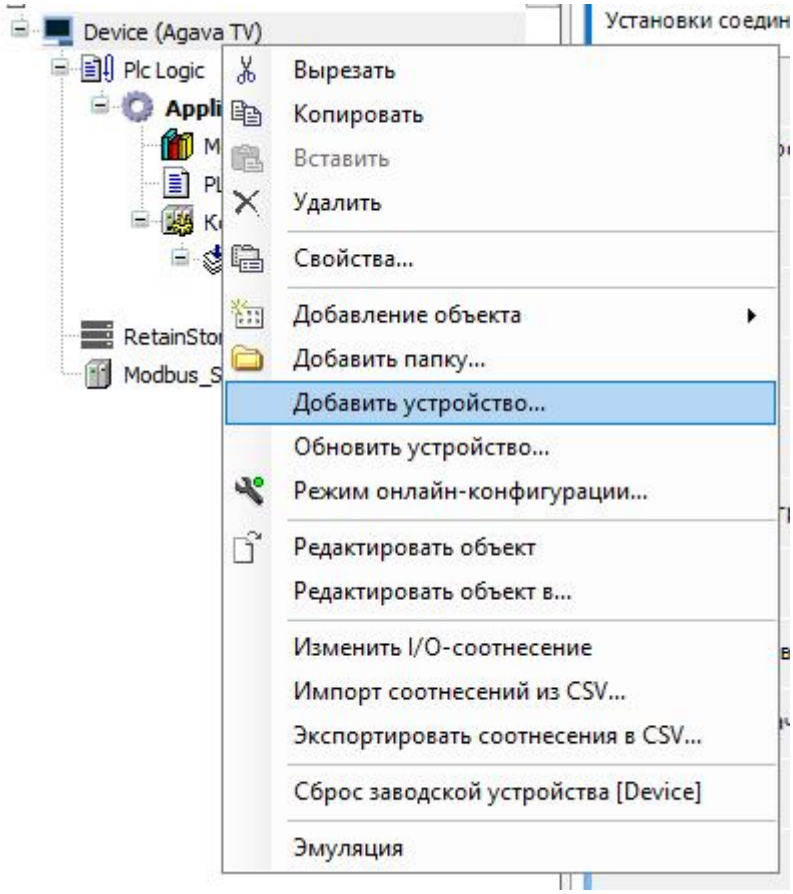


Рисунок 5. Добавление нового устройства

В открывшемся окне выбираем группу «**Разн.**», далее выбираем пункт из списка **ModulesPLC-40**, нажимаем кнопку «**Добавить устройство**». После того как устройство добавлено в дерево проекта, нажимаем кнопку «**Закрыть**».

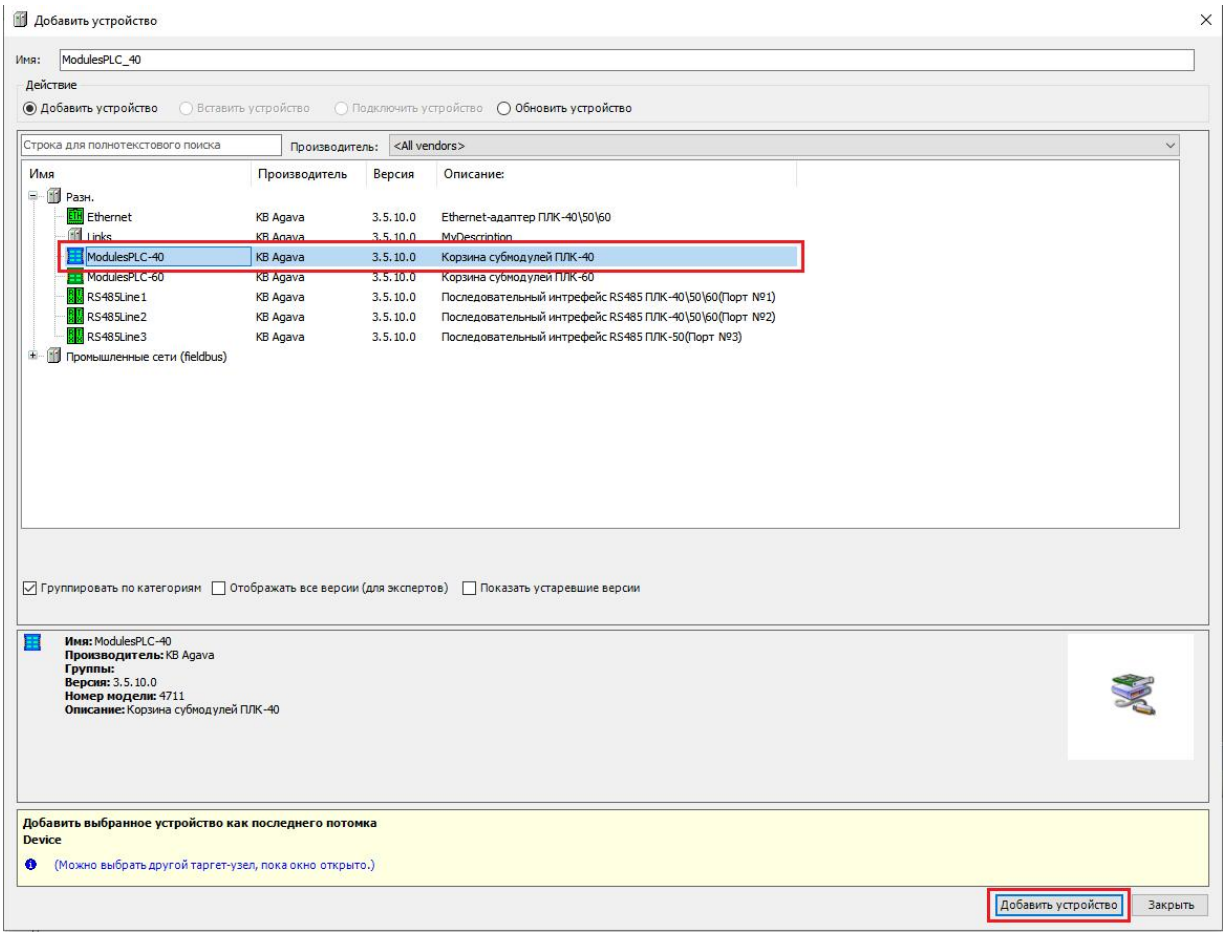


Рисунок 5-1. Добавление нового устройства

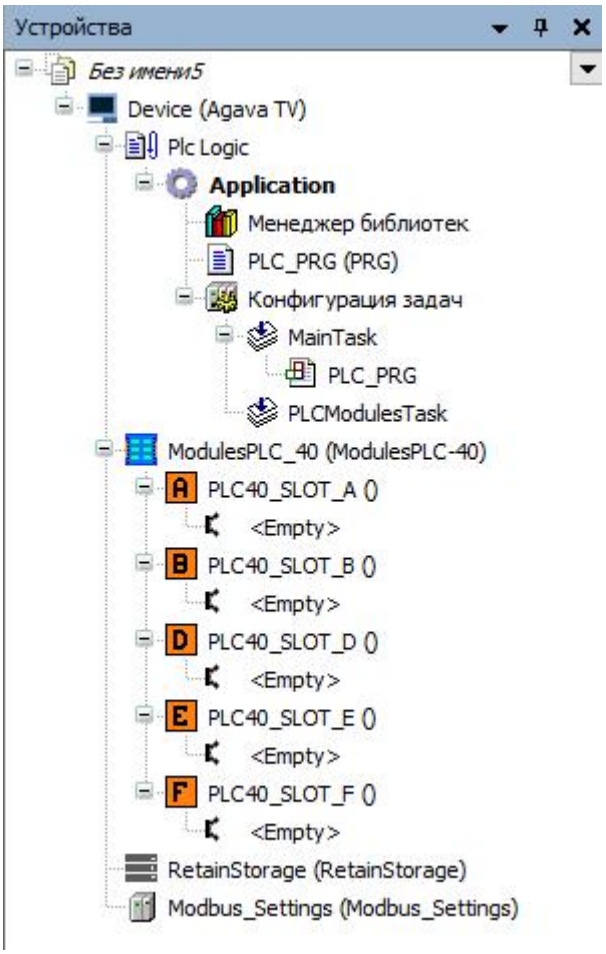


Рисунок 5-2. Добавление нового устройства

После того как корзина субмодулей ПЛК-40 добавлена в дерево проекта, можно приступать к конфигурации субмодулей.

Текущее расположение субмодулей в корзине ПЛК-40, в тестовой конфигурации имеет следующий порядок:

СЛОТ А - Субмодуль **DI**
СЛОТ В - Субмодуль **R**
СЛОТ С - Блок питания
СЛОТ D - Пустой слот
СЛОТ E - Субмодуль **TMP**
СЛОТ F - Субмодуль **AI**

Для подключения субмодуля к слоту в корзине ПЛК-40 кликнем правой клавишей мыши по слоту, в открывшемся контекстном меню выберем пункт «**Подключить устройство**».

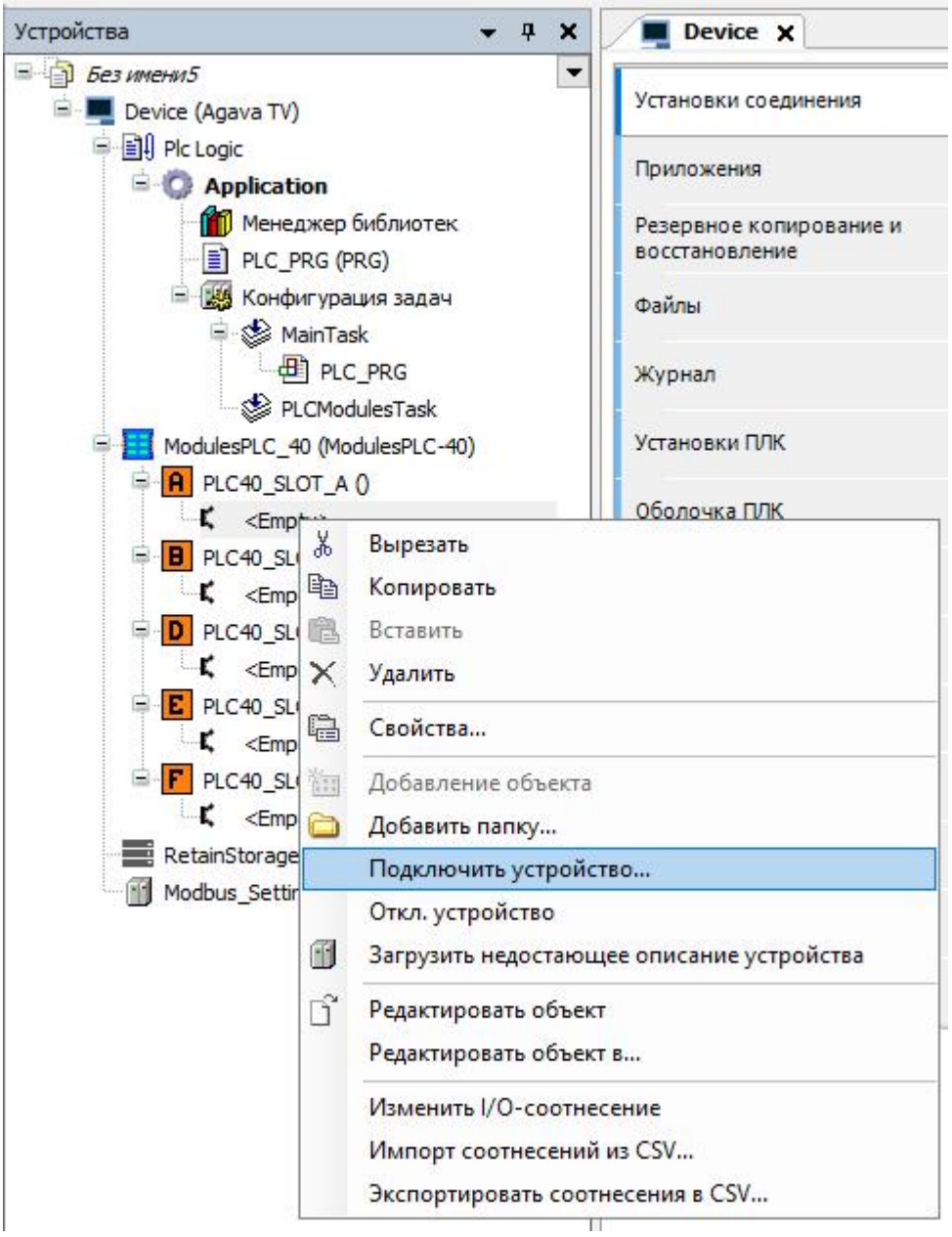


Рисунок 5-3. Подключение субмодуля

В открывшемся окне укажем нужный тип субмодуля, в нашем случае это **DI**, далее двойным кликом мыши подключим субмодуль. Аналогичным способом не закрывая окно, укажем нужные субмодули и подключим к слотам.

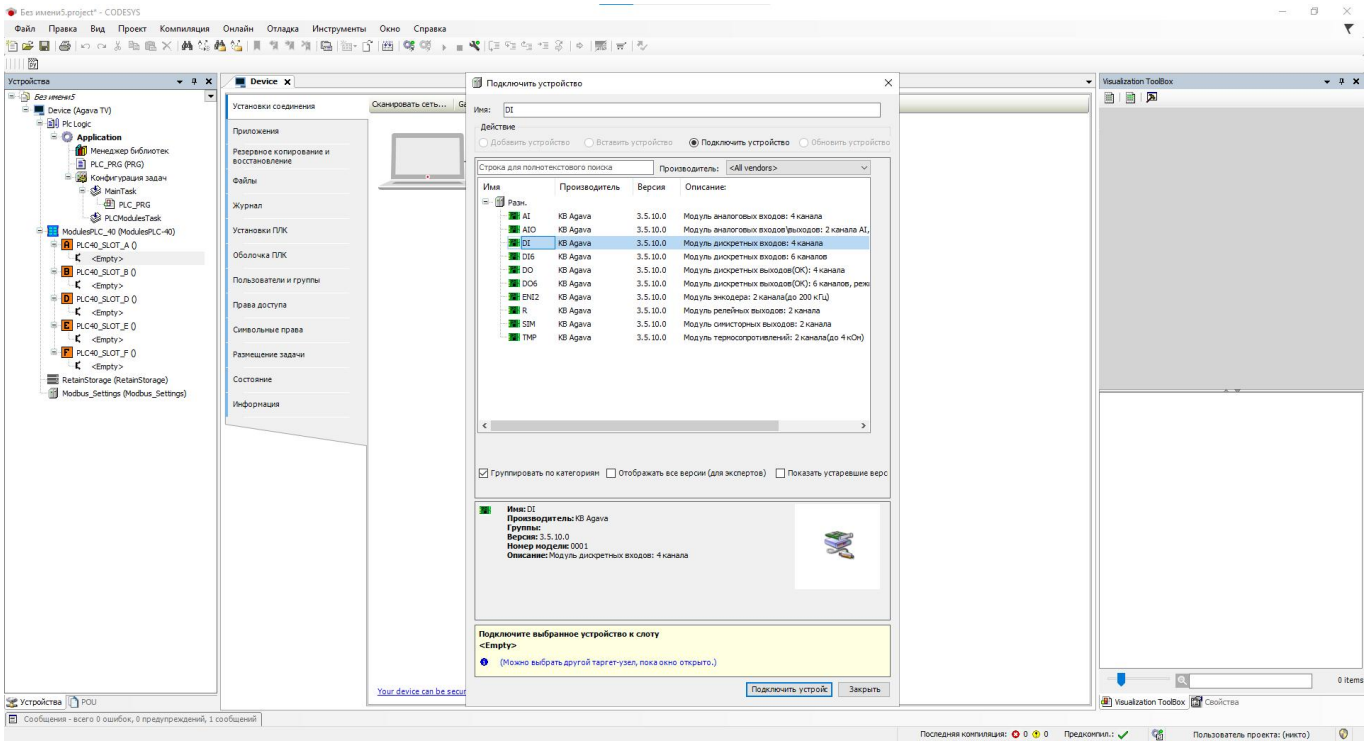


Рисунок 5-4. Подключение субмодуля

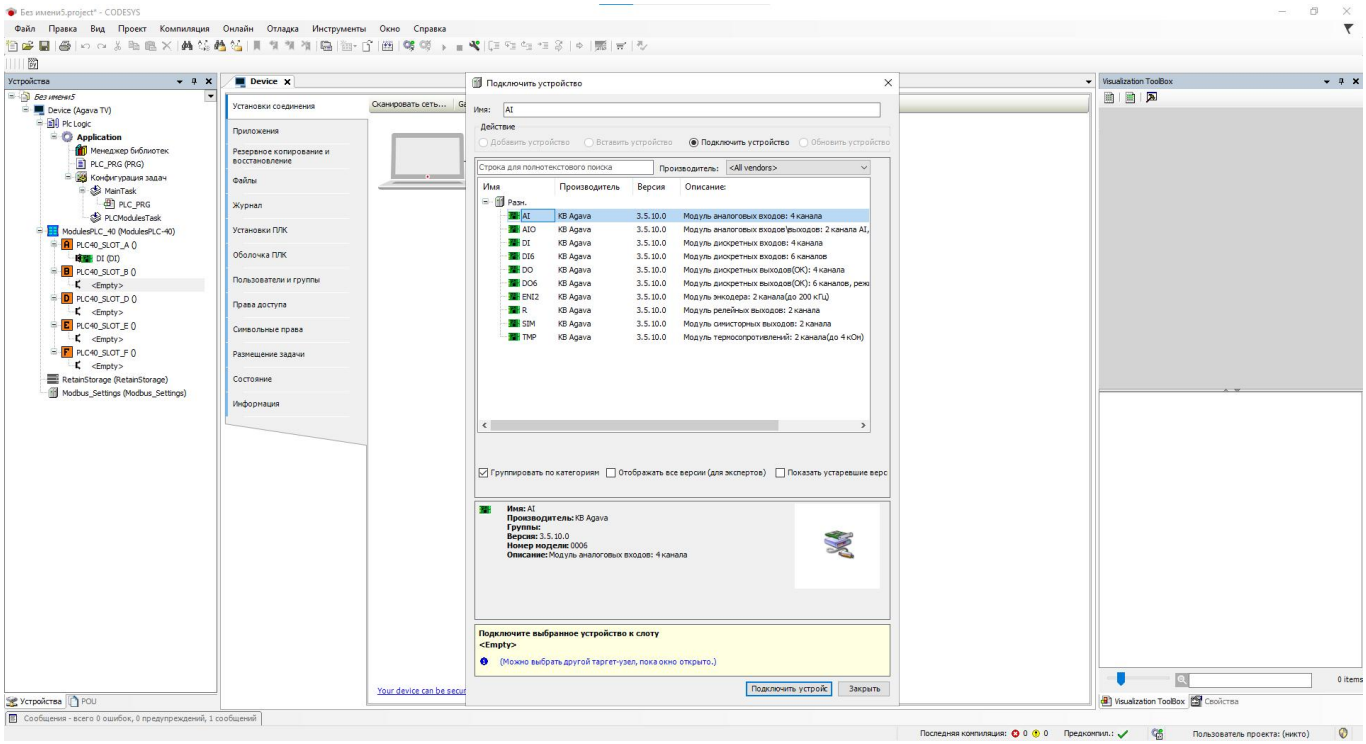


Рисунок 5-5. Подключение субмодуля

После завершения конфигурации закроем окно по нажатию кнопки «**Закрыть**».

После того как формирование корзины ПЛК-40 завершено, можно приступать к настройке субмодулей.

Для настройки субмодуля **DI** дважды кликнем по нему левой клавишей мыши и выберем пункт «**Internal Соотнесение входов/выходов**».

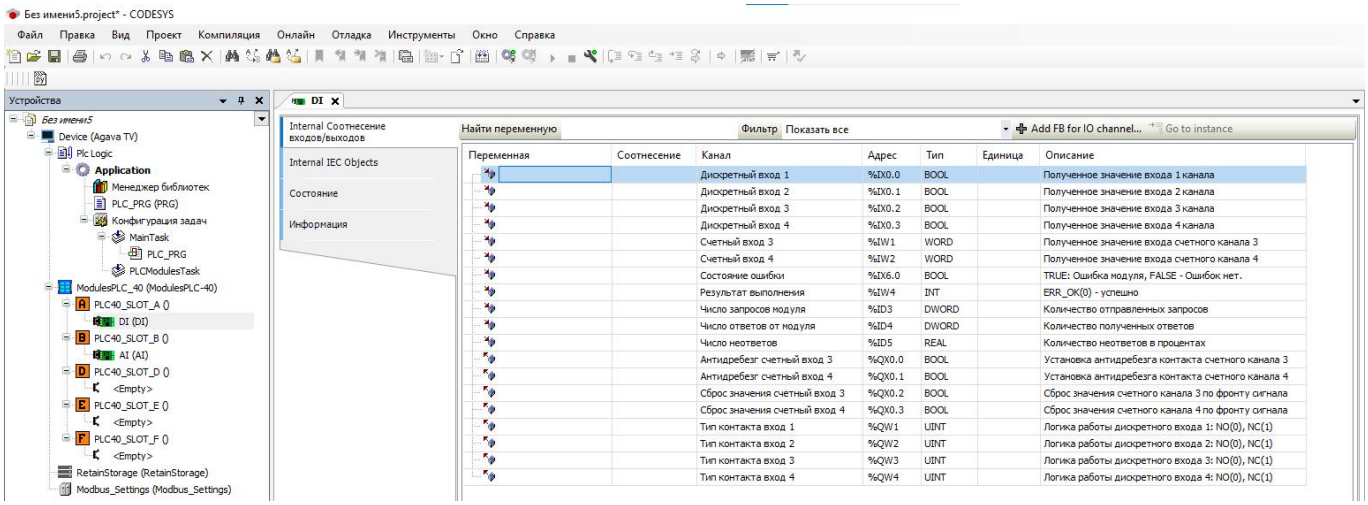


Рисунок 6. Настройка субмодуля DI

Столбец «**Переменная**» позволяет создать соотнесение переменной проекта с каналом субмодуля. В качестве теста можно создать переменную нужного типа и сделать соотнесение с каналом субмодуля или использовать готовую структуру сигнала **TSensorsStruct**. В данном примере будет рассмотрено использование структуры **TSensorsStruct**.

```
// Структура описания аналогового/дискретного датчика
type TSensorsStruct :
struct

    id:             uint;           // Идентификатор сигнала
    Name:           wstring;        // Имя сигнала
    ShortName :     wstring;        // Краткое обозначение
    Unit:           wstring;        // Размерность
    bError:         bool;           // Признак наличия ошибки обмена
    channel:        wstring;        // Описание канала. Например( ПЛК40.A.X.1.1)
    AiType:         EnSensTypeAI;   // Тип сигнала AI
    TMPType:        EnSensTypeTMP;  // Тип сигнала TMP
    LogicType:      EnLogicType;    // Тип сигнала DI\DO
    MaxLimValue:    real;           // Верхний предел измерения сигнала
    MinLimValue:    real;           // Нижний предел измерения сигнала
    ErrorId:        uint;           // Код ошибки сигнала
    rValue:         real;           // Значение сигнала(float)
    bValue:         bool;           // Значение сигнала(bool)
    tLpf:           int;            // Время фильтра ФНЧ(мс)
    Cnt3 :          uint;           // Счетный вход канала №3 субмодуля DI
    Cnt4 :          uint;           // Счетный вход канала №4 субмодуля DI
    Deb3 :          bool;           // Флаг установки антидребезга счетного канала №3 субмодуля DI
    Deb4 :          bool;           // Флаг установки антидребезга счетного канала №4 субмодуля DI
    ResCnt3 :       bool;           // Сброс счетного счетного канала №3 субмодуля DI
    ResCnt4 :       bool;           // Сброс счетного счетного канала №4 субмодуля DI

end_struct
end_type
```

Рисунок 7. Описание структуры TSensorsStruct

Объявим необходимые экземпляры структур **TSensorsStruct** в программе **PLC_PRG**.

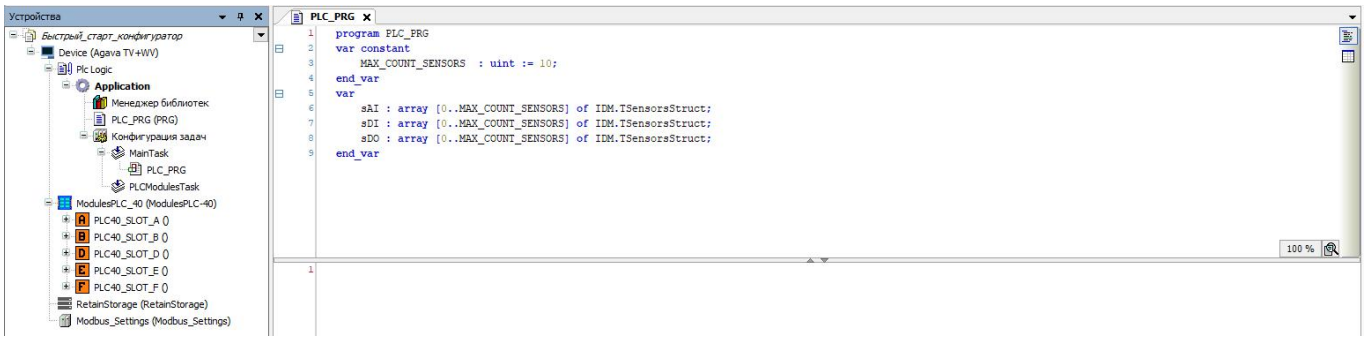


Рисунок 8. Объявление экземпляров структуры TSensorsStruct в PLC_PRG

В окне настроек сумодуля **DI** выполним соотнесение структуры с каналами. Для этого кликнем левой клавишей мыши в поле столбца «**Переменная**» и нажмём кнопку  , в открывшемся окне выберем **Application -> sDI -> bValue** и нажмем **ОК**.

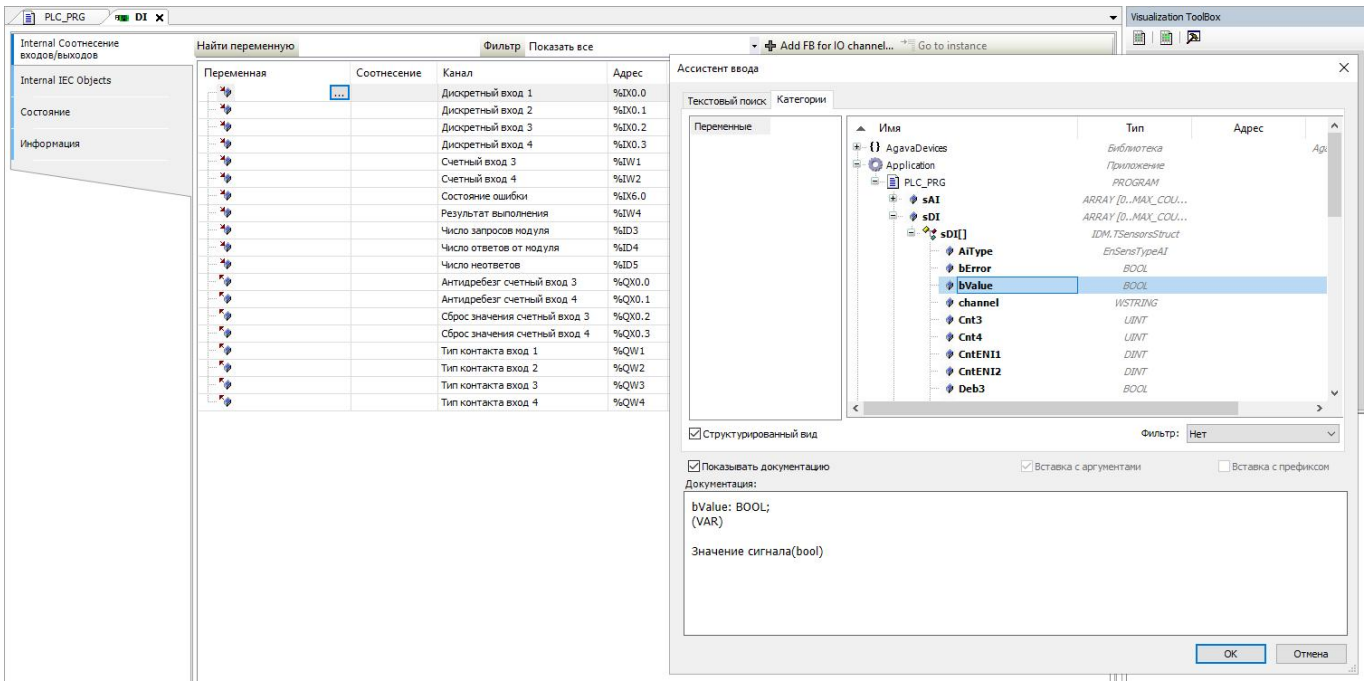


Рисунок 9. Соотнесение переменной структуры с каналом модуля

Так как объявленные переменные представлены массивом, указываем соответствующий индекс нужного элемента массива.

Переменная	Соотнесение	Канал	Адрес	Тип	Единица	Описание
Application.PLC_PRG.sDI[0].bValue		Дискретный вход 1	~~%IX0.0~~	BOOL		Полученное значение входа 1 канала
Application.PLC_PRG.sDI[1].bValue		Дискретный вход 2	~~%IX0.1~~	BOOL		Полученное значение входа 2 канала
Application.PLC_PRG.sDI[2].bValue		Дискретный вход 3	~~%IX0.2~~	BOOL		Полученное значение входа 3 канала
Application.PLC_PRG.sDI[3].bValue		Дискретный вход 4	~~%IX0.3~~	BOOL		Полученное значение входа 4 канала
		Счетный вход 3	%IW1	WORD		Полученное значение входа счетного канал
		Счетный вход 4	%IW2	WORD		Полученное значение входа счетного канал
		Состояние ошибки	%IX6.0	BOOL		TRUE: Ошибка модуля, FALSE - Ошибок нет.
		Результат выполнения	%IW4	INT		ERR_OK(0) - успешно
		Число запросов модуля	%ID3	DWORD		Количество отправленных запросов
		Число ответов от модуля	%ID4	DWORD		Количество полученных ответов
		Число неответов	%ID5	REAL		Количество неответов в процентах
		Антидребезг счетный вход 3	%QX0.0	BOOL		Установка антидребезга контакта счетного
		Антидребезг счетный вход 4	%QX0.1	BOOL		Установка антидребезга контакта счетного
		Сброс значения счетный вход 3	%QX0.2	BOOL		Сброс значения счетного канала 3 по фронт
		Сброс значения счетный вход 4	%QX0.3	BOOL		Сброс значения счетного канала 4 по фронт
		Тип контакта вход 1	%QW1	UINT		Логика работы дискретного входа 1: NO(0),
		Тип контакта вход 2	%QW2	UINT		Логика работы дискретного входа 2: NO(0),
		Тип контакта вход 3	%QW3	UINT		Логика работы дискретного входа 3: NO(0),
		Тип контакта вход 4	%QW4	UINT		Логика работы дискретного входа 4: NO(0),

Рисунок 9-1. Соотнесение переменной структуры с каналом модуля

Компилируем (F11), загружаем (Alt+F8) и выполняем проект (F5), зелёные индикаторы на против субмодуля сигнализируют об успешном обмене, статистика обмена отображает текущее значение отправленных и полученных данных, а также количество ошибок(число неответов в процентном соотношении) .

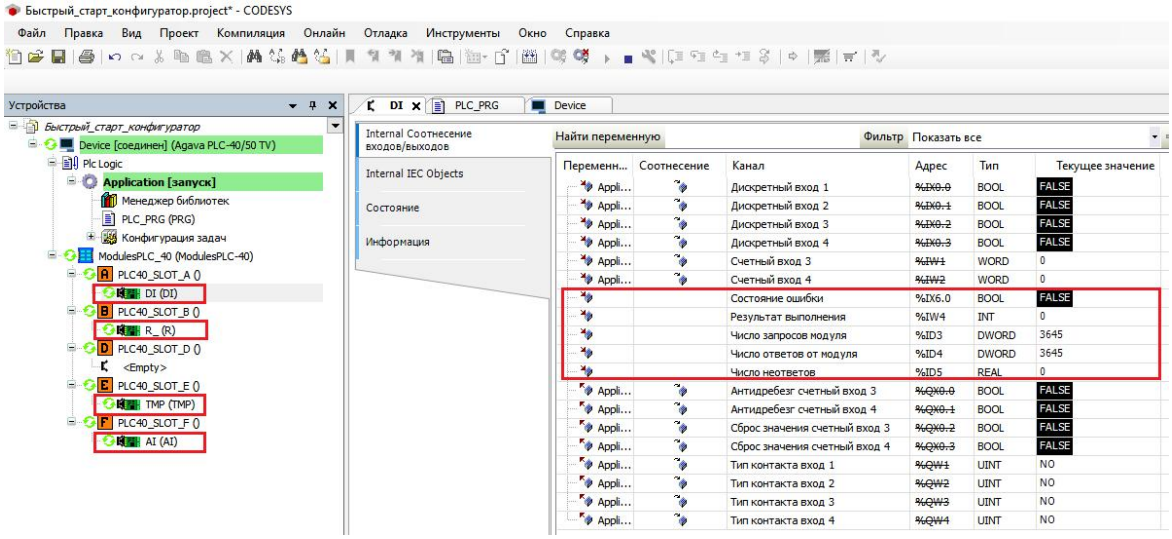


Рисунок 10-1. Соотнесение переменной структуры с каналом модуля

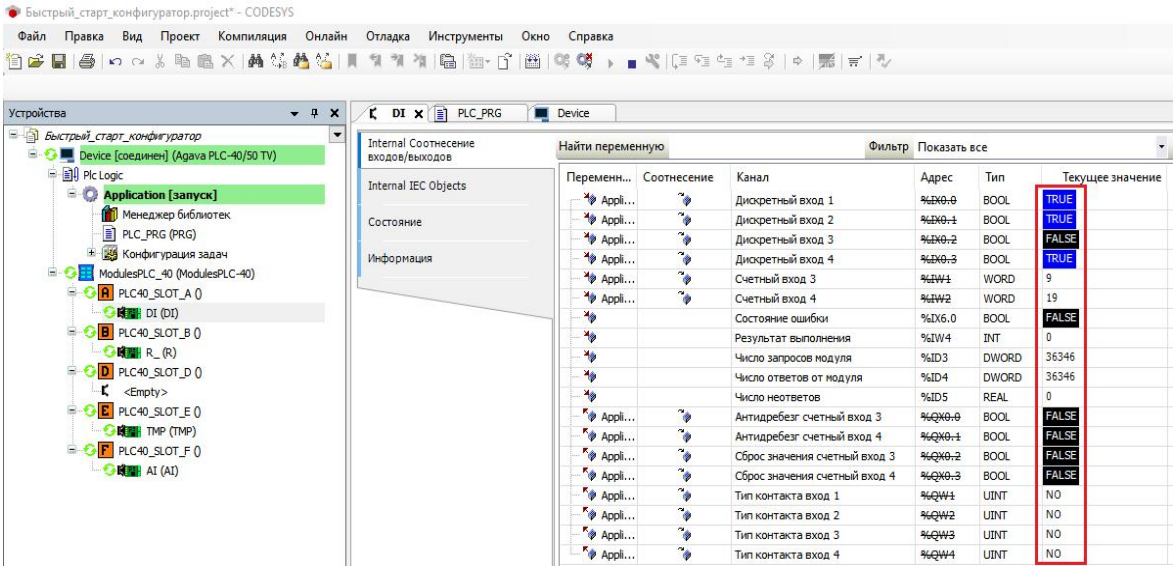


Рисунок 10-2. Соотнесение переменной структуры с каналом модуля

При замыкании дискретного входа мы видим, что сигнал получен, а соотнесение передаст значение сигнала переменной в проекте.

Индикация обмена на против каждого субмодуля позволяет оперативно определить проблему работы субмодуля, например если пользователь перепутал расположение субмодулей или установил в корзину не существующий субмодуль, то напротив проблемного субмодуля будет изображён красный треугольник.

На рисунке 11 показан пример ошибочной установки субмодулей в слоты A и B.

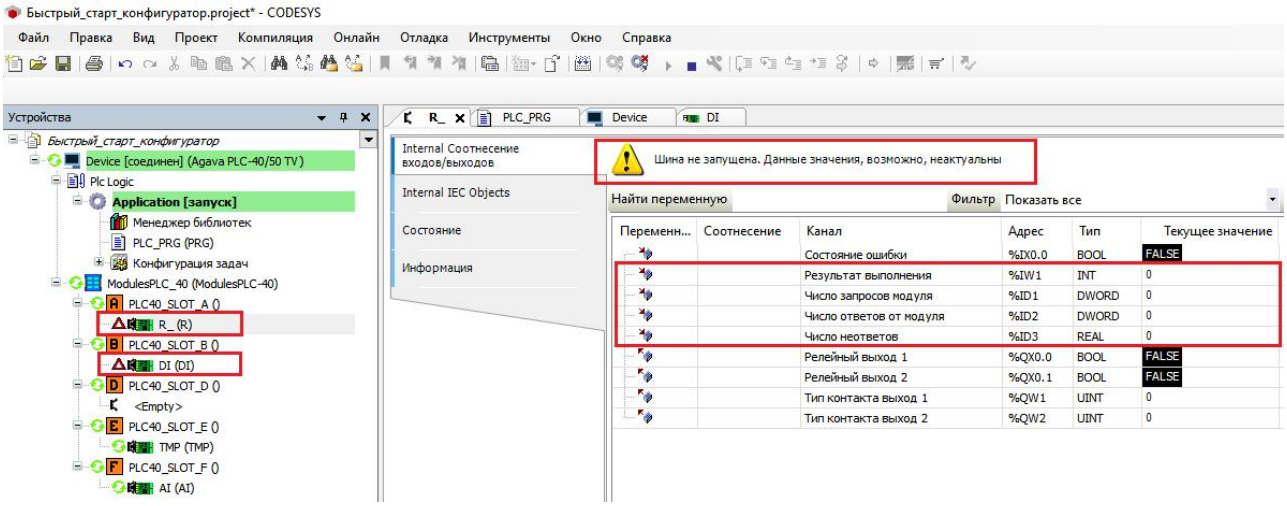


Рисунок 11. Индикация ошибки обмена или некорректной установки субмодуля

Для управления релейными выходами достаточно реализовать соотнесение переменной с каналами субмодуля **R**.

Для проверки исправности работы выходов без соотнесения переменной, в столбец «**Подготовленное значение**» требуется установить значение **TRUE**, для этого этого необходимо кликнуть левой клавишей мыши в поле столбца на против канала модуля и нажать (Ctrl+F7).

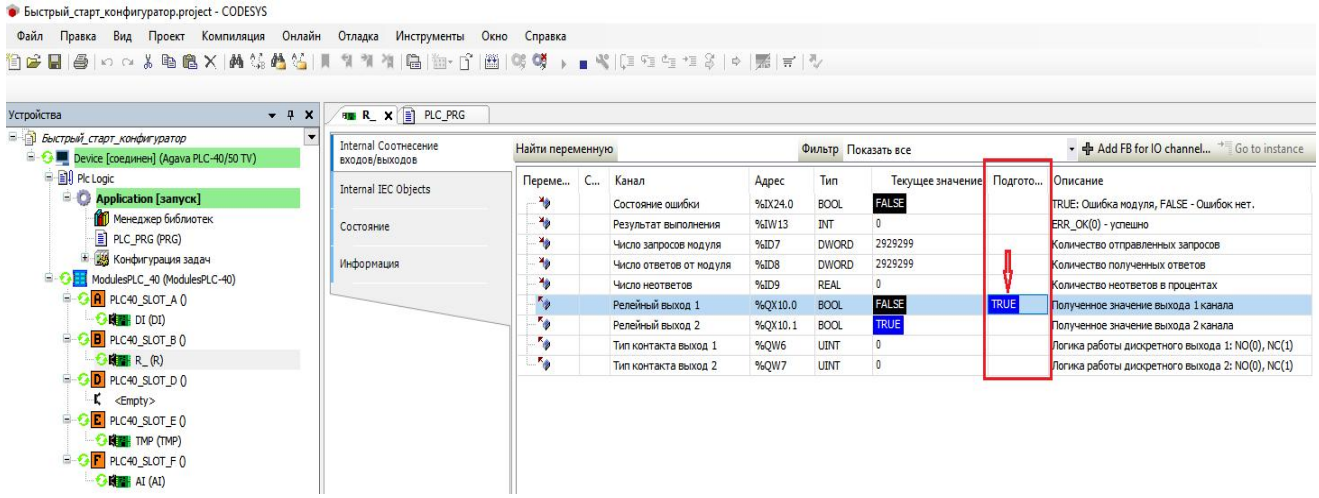


Рисунок 12. Проверка срабатывания релейного выхода

При необходимости можно реализовать инверсную логику срабатывания выхода, для этого в поле «**Тип контакта выхода 1**» нужно установить значение 1(NC) , тогда релейный выход будет работать инверсно и по умолчанию будет иметь замкнутый контакт. Аналогичная настройка также имеется у субмодуля **DI**.

Субмодуль **TMP** по умолчанию отображает значение в **Ом**, в качестве теста к каналам модуля подключены сопротивления номиналом 61.4 и 66.6 **Ом**, в полях «**Тип термосопротивления входа 1**» и «**Тип термосопротивления входа 2**» указывается тип датчика. После установки нужного типа датчика, значение канала будет отображать температуру в градусах цельсия.

Описание типов датчиков представлено в разделе **7.3.23. Перечисления**.

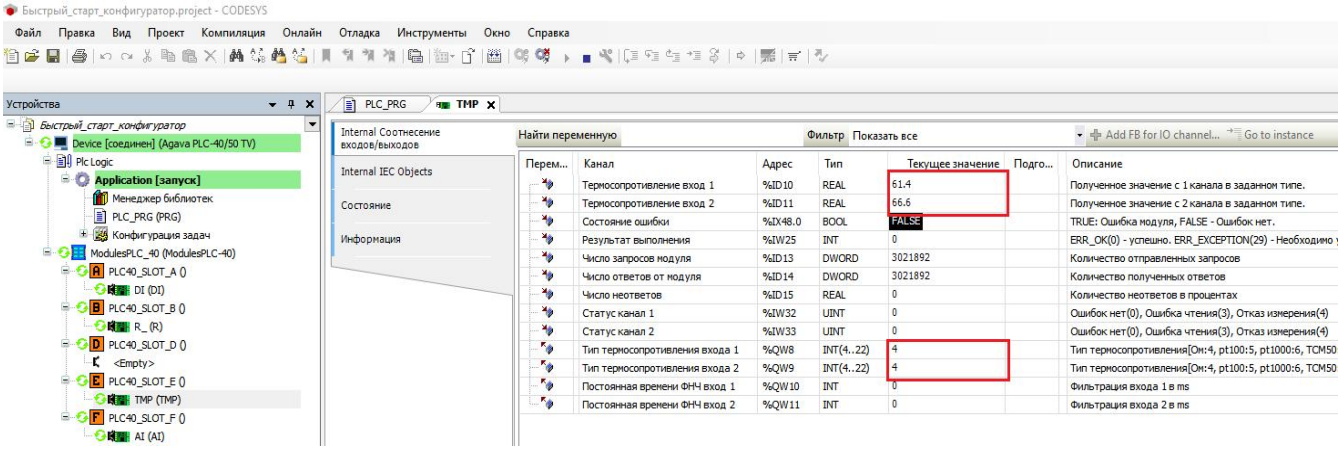


Рисунок 13. Проверка чтения каналов субмодуля TMP

Субмодуль **AI** имеет четыре универсальных конфигурируемых канала, рассмотрим вариант настройки датчиков помощью структуры **TSensorsStruct**.

В программу **PLC_PRG** добавим метод инициализации датчиков, для этого кликнем **ПКМ** по программе **PLC_PRG** из контекстного меню выберем пункт «**Добавление объекта**», далее «**Метод**», зададим название метода **InitSensors**, возвращаемое значение оставим пустым и нажмём кнопку «**Добавить**». В программе объявим новую переменную **bInitSensors**.

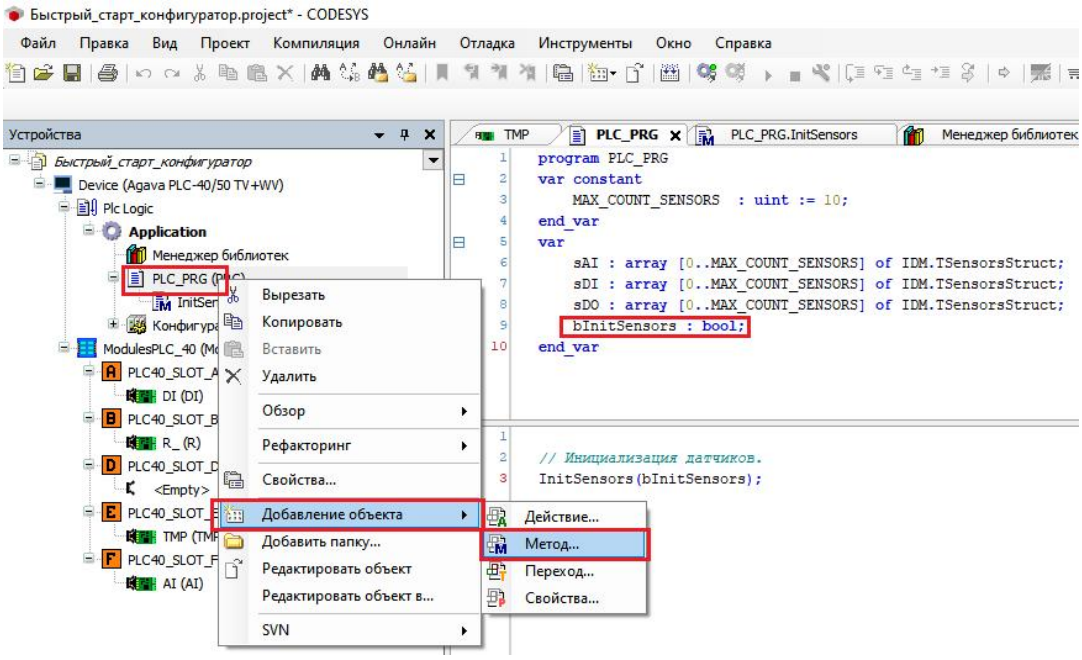


Рисунок 14. Добавление метода

Опишем реализацию метода как показано на рисунке ниже.

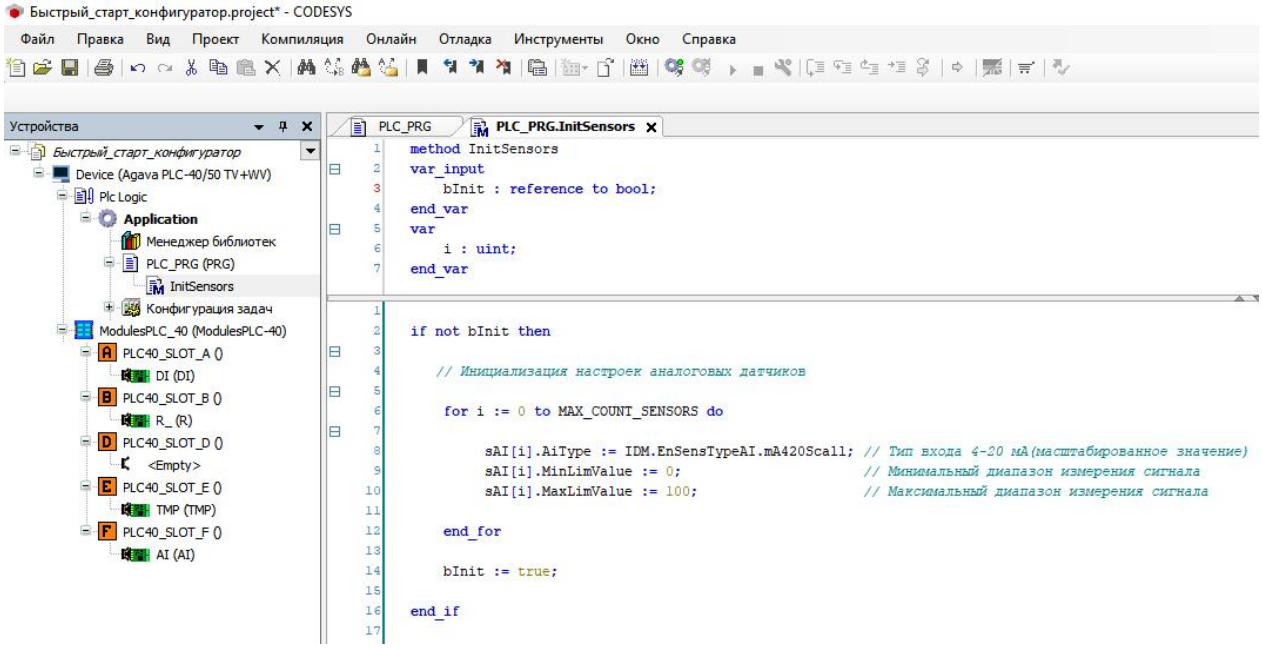


Рисунок 15. Описание реализации метода

Добавим вызов метода в основной программе **PLC_PRG**.

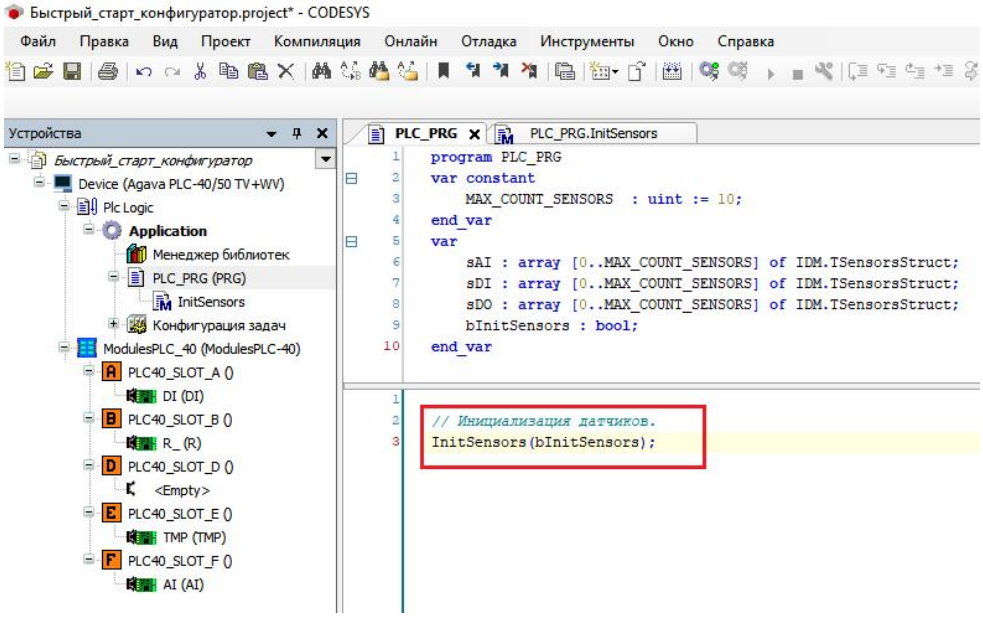


Рисунок 16. Добавление вызова метода в основной программе

Далее производим соотнесение переменных структуры с каналами субмодуля **AI**, компилируем (F11), загружаем (Alt+F8) и выполняем проект (F5).

Первый канал субмодуля считывает тестовое значение задатчика тока равное **20 мА**, тип значения автоматически пересчитывается по установленному диапазону датчика, мы можем увидеть полученное значение **100**.

Статус первого канала отображает код шибки равный нулю, что свидетельствует об успешном чтении значения сигнала. Расшифровка кодов ошибок каналов представлена в столбце «**Описание**».

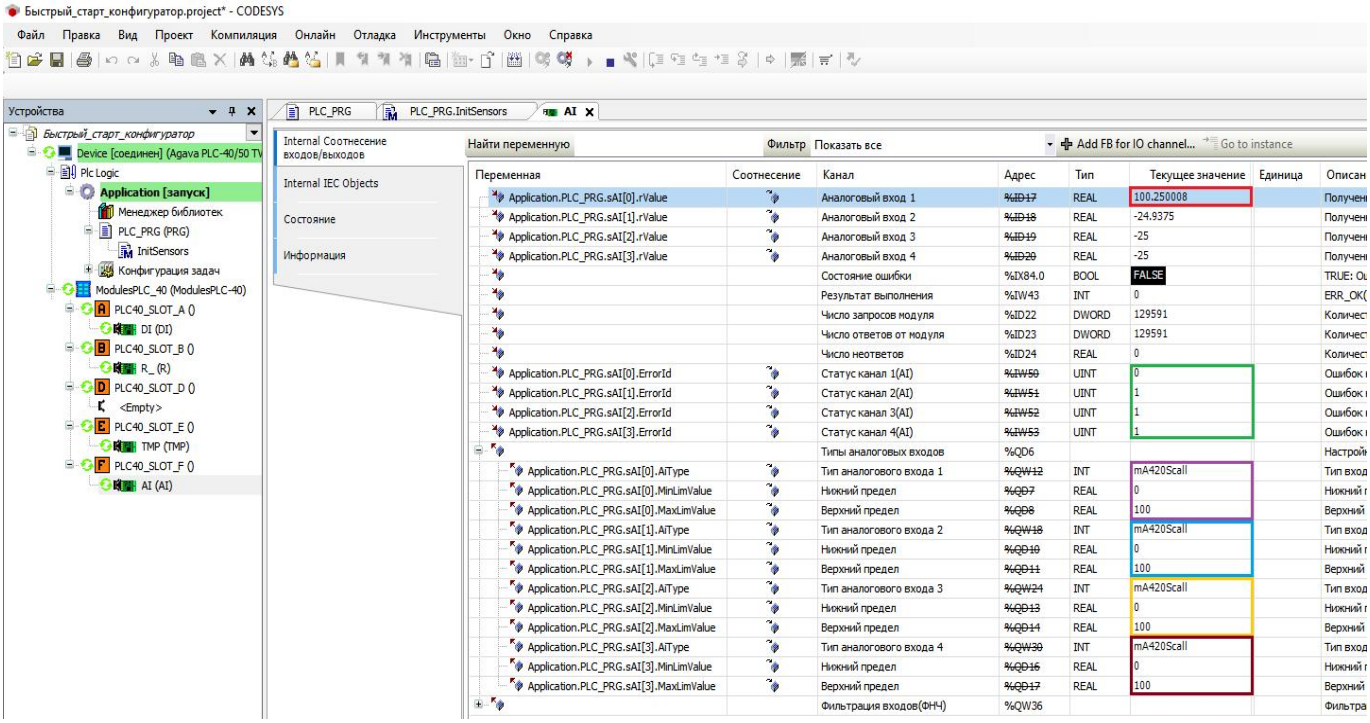


Рисунок 17. Проверка чтения каналов субмодуля AI

5.2. Проект с использованием конфигуратора МВВ-40

5.2.1. МВВ-40 модификации 1 и 2

Рассмотрим пример добавления модулей **МВВ-40 модификации 1 и 2**. Способ подключения и конфигурации модулей расширения, рассмотренных в данном примере, может быть использован для серии контроллеров: **АГАВА ПЛК-40**, **АГАВА ПЛК-50**, **АГАВА ПЛК-60**.

В качестве примера используется контроллер **ПЛК-40** с установленными субмодулем интерфейса 485 , место расположение субмодуля в корзине **ПЛК-40** значение не имеет, а также модуль расширения **МВВ-40.2** (для модификации 1 способ подключения аналогичный).

Для **МВВ-40.2** требуется первоначальная настройка сетевого адреса, которую можно произвести с помощью утилиты «**Конфигуратор МВВ-40**». Утилиту можно скачать с официального сайта компании по ссылке: https://www.kb-agava.ru/index.php?route=module/product_downloads/get&did=342.

В окне утилиты переходим на вкладку **«Последовательный порт»,** указываем порт преобразователя интерфейса USB/RS485 и нажимаем кнопку **«Подключиться»**, начнётся сканирование модулей, найденные модули будут отображены в списке.

Нажимаем кнопку **«Конфигурирование модуля»**

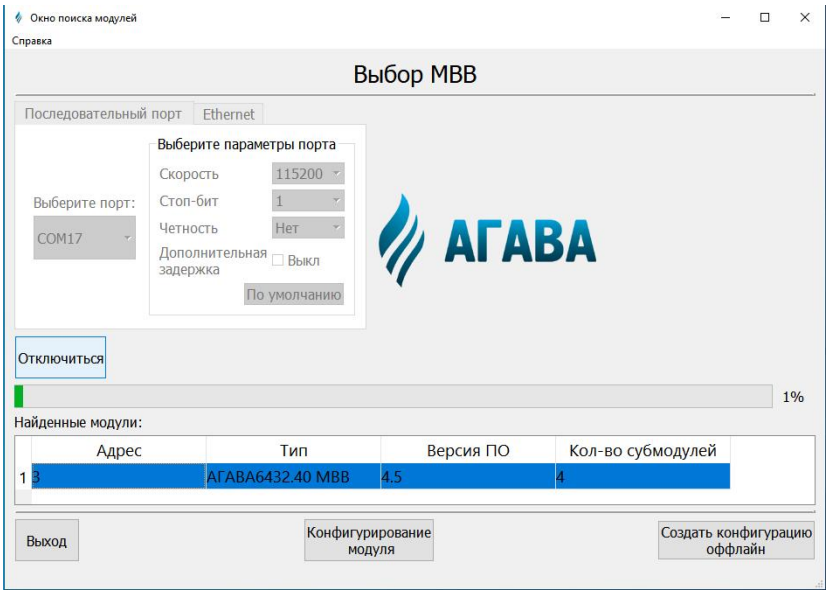


Рисунок 18. Конфигуратор МВВ-40

В открывшемся окне настраиваем параметры связи:

Адрес MODBUS: 1

Скорость, бит/с: 115200

Чётность, стоп-биты: нет / 1 стоп-бит

Нажимаем кнопку **«Записать изменённые параметры»**. В окне конфигуратора также можно увидеть наличие субмодулей установленных в корзине **МВВ-40.2** и их тип, дополнительной настройки субмодулей не требуется. Более подробно об утилите конфигуратора МВВ-40 рассказано в **Агава МВВ-40 РЭ** https://www.kb-agava.ru/index.php?route=module/product_downloads/get&did=341

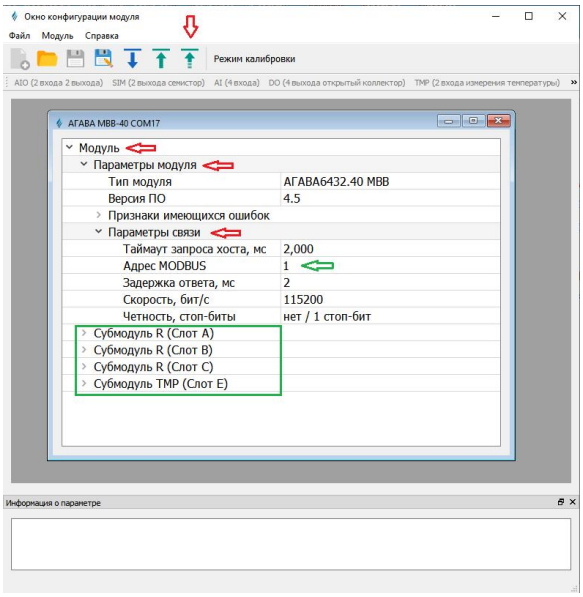


Рисунок 18-1. Конфигуратор МВВ-40

После того как модуль расширения настроен, его необходимо подключить к интерфейсу RS485 **ПЛК-40**.

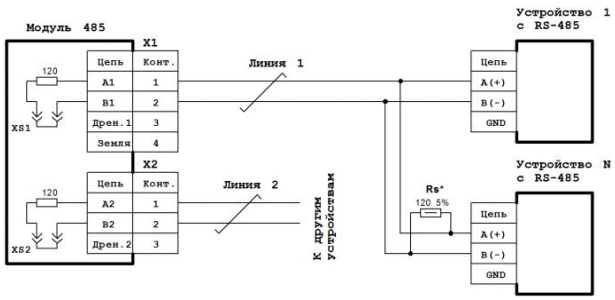


Рисунок 19. Схема подключения субмодуля 485 к линии RS-485

В данном примере будет рассмотрено использование 1 канала субмодуля 485 (**Порт №1**).

Создадим новый проект, выберем описание нужного типа устройства, настроим опцию **«Всегда обновлять переменные»** в установках ПЛК, аналогично способу рассмотренному в разделе **5.1.**

Кликнем правой кнопкой мыши по **Device** и выберем пункт меню «**Добавить устройство**», в открывшемся окне выделим **«RS485Line1»** и нажмём кнопку **«Добавить устройство»**, а затем кнопку **«Закрыть»**.

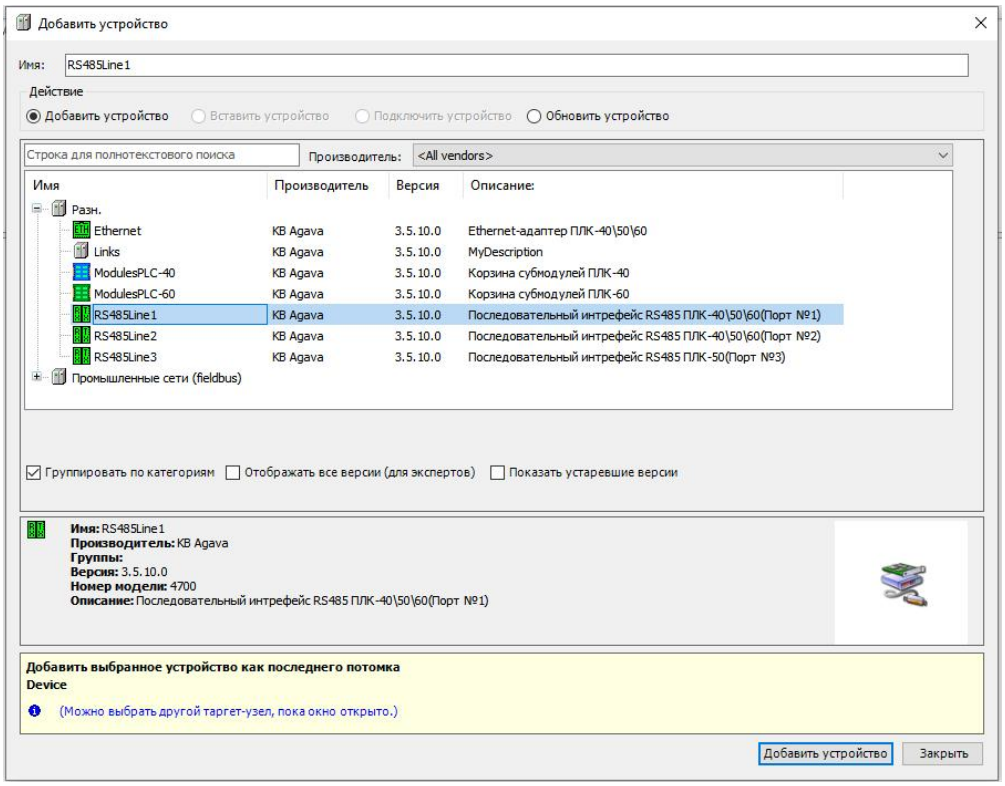


Рисунок 20. Добавление интерфейса RS485Line1

После добавления интерфейса появляется возможность его настройки, в данном примере оставляем настройки по умолчанию.

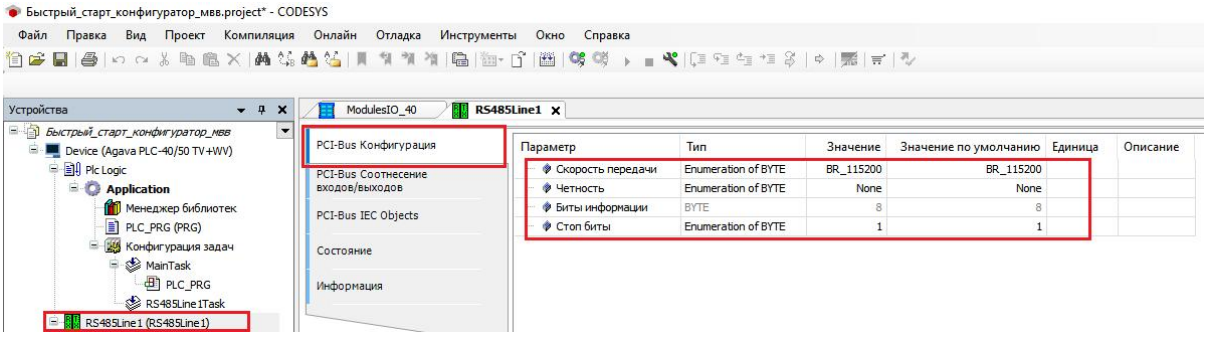


Рисунок 21. Настройка интерфейса RS485Line1

В дереве проекта кликнем правой клавишей мыши по **RS485Line1,** в контекстном меню выберем пункт **«Добавить устройство»**, в открывшемся окне выделим **ModulesIO-40** и нажмём кнопку **«Добавить устройство»,** закроем окно по нажатию кнопки **«Закрыть»**.

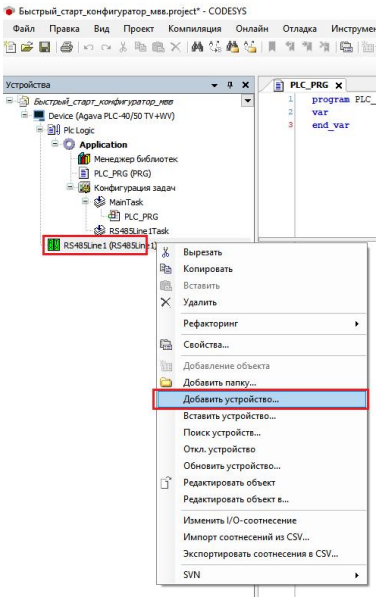


Рисунок 22. Добавление нового устройства

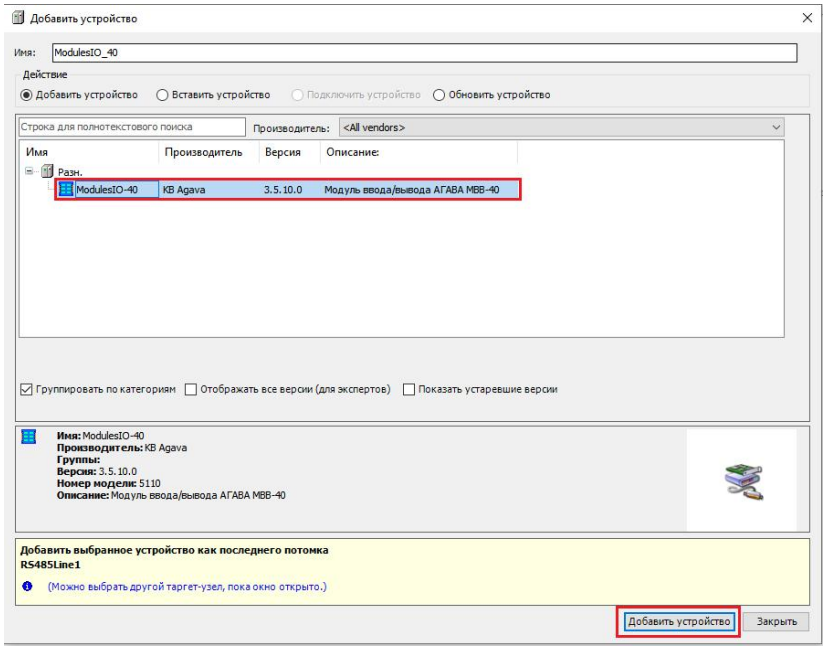


Рисунок 22-1. Добавление нового устройства

После того как модуль МВВ-40 добавлен в дерево проекта, ему можно назначить адрес устройства. Для настройки нужно дважды кликнуть левой клавишей мыши по **ModulesIO_40**. В нашем случае модулю установлен адрес устройства **1**, поэтому настройки оставляем по умолчанию.

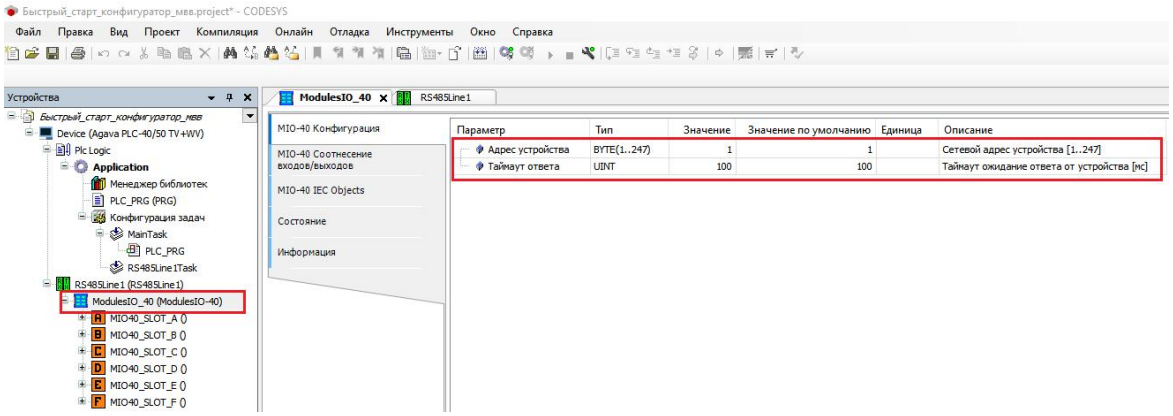


Рисунок 23. Настройка адреса устройства

Далее подключаем необходимые субмодули к слотам корзины МВВ-40, аналогично способу рассмотренному в разделе **5.1**

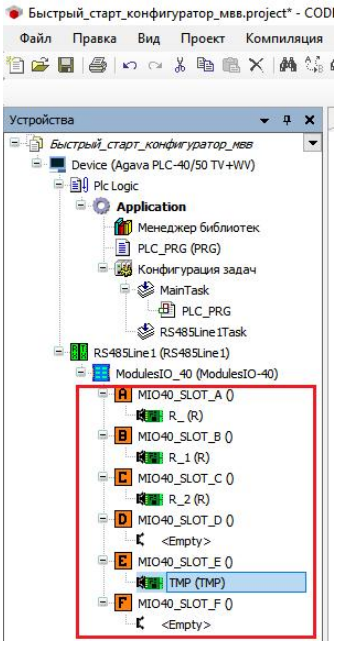


Рисунок 24. Подключение субмодулей к слотам корзины МВВ-40

Компилируем (F11), загружаем (Alt+F8) и выполняем проект (F5).

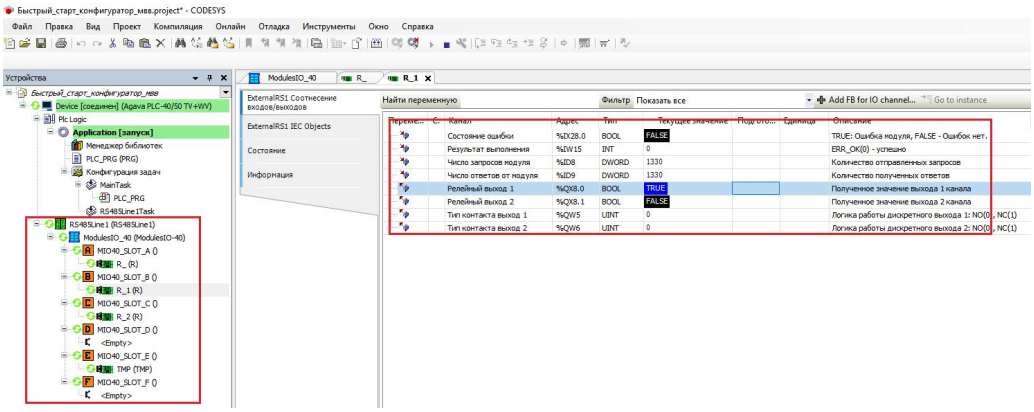


Рисунок 24. Выполнение проекта

Дополнительные модули расширения добавляются в проект аналогичными способом, кликом правой клавиши мыши по **RS485Line1 -> Добавить устройство.** Каждому новому устройству соответственно назначается свой сетевой адрес.

5.2.2. МВВ-40 модификации 3

Рассмотрим пример добавления модулей **МВВ-40 модификации 3**. Способ подключения и конфигурации модулей расширения, рассмотренных в данном примере, может быть использован для серии контроллеров: **АГАВА ПЛК-40**, **АГАВА ПЛК-50**, **АГАВА ПЛК-60**.

В качестве примера используется контроллер **ПЛК-40** с установленными субмодулем интерфейса RS232/ETH, **место расположение субмодуля в корзине ПЛК-40 - Слот D**.

Для **МВВ-40.3** требуется первоначальная настройка IP адреса, которую можно произвести с помощью утилиты «**Конфигуратор МВВ-40**». Утилиту можно скачать с официального сайта компании по ссылке: https://www.kb-agava.ru/index.php?route=module/product_downloads/get&did=342.

В окне утилиты выбираем вкладку **«Ethernet»** и нажимаем кнопку **«Подключиться».** По умолчанию модуль МВВ-40.3 имеет IP адрес 192.168.10.130.

Нажимаем кнопку **«Конфигурирование модуля»**

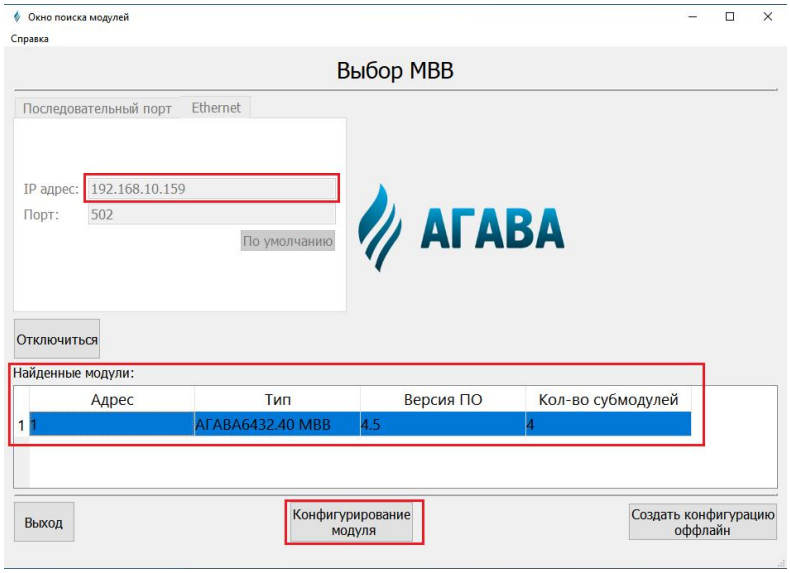


Рисунок 25. Конфигуратор МВВ-40

В открывшемся окне настраиваем параметры связи и нажимаем кнопку **«Записать изменённые параметры»**. В окне конфигуратора также можно увидеть наличие субмодулей установленных в корзине **МВВ-40.3** и их тип, дополнительной настройки субмодулей не требуется. Более подробно об утилите конфигуратора МВВ-40 рассказано в **Агава МВВ-40 РЭ** https://www.kb-agava.ru/index.php?route=module/product_downloads/get&did=341

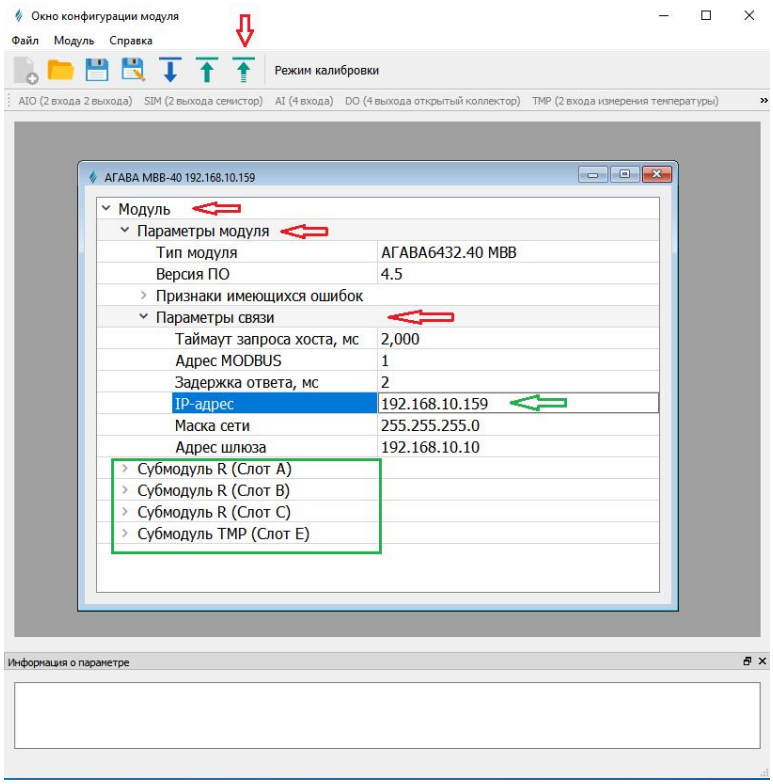


Рисунок 26. Конфигуратор МВВ-40

Создадим новый проект, выберем описание нужного типа устройства, настроим опцию **«Всегда обновлять переменные»** в установках ПЛК, аналогично способу рассмотренному в разделе **5.1.**

Кликнем правой клавишей мыши по **Device** и выберем пункт меню «**Добавить устройство**», в открывшемся окне выделим **«Ethernet»** и нажмём кнопку **«Добавить устройство»**, а затем кнопку **«Закрыть»**.

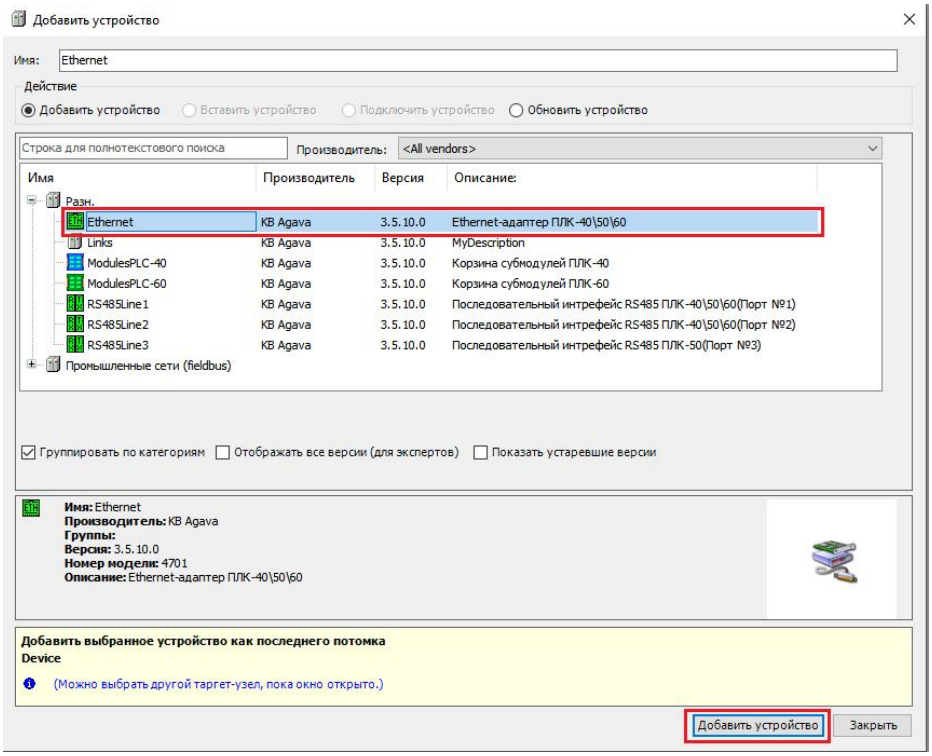


Рисунок 27. Добавление интерфейса Ethernet

В дереве проекта кликнем правой клавишей мыши по **Ethernet**, в контекстном меню выберем пункт **«Добавить устройство»**, в открывшемся окне выделим **ModulesIO-40** и нажмём кнопку **«Добавить устройство»,** закроем окно по нажатию кнопки **«Закрыть»**.

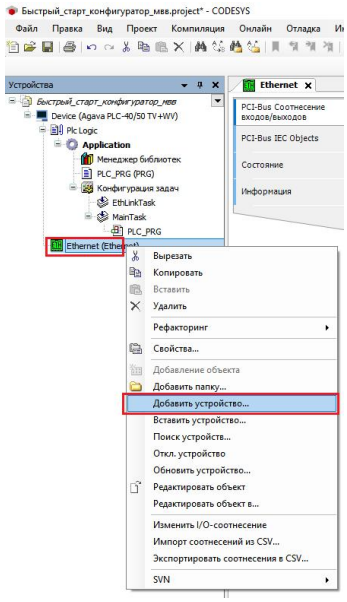


Рисунок 28. Добавление нового устройства

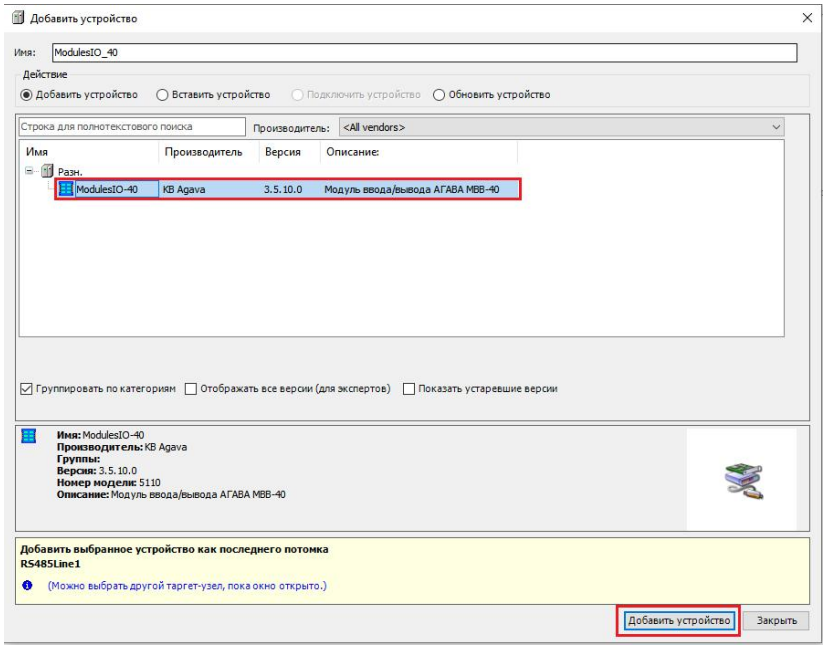


Рисунок 28-1. Добавление нового устройства

После того как модуль МВВ-40 добавлен в дерево проекта, ему можно задать IP адрес. Для настройки нужно дважды кликнуть левой клавишей мыши по **ModulesIO_40,** далее двойными кликом левой клавиши мыши в столбце **«Значение»** записать IP адрес устройства, в нашем случае это 192.168.10.159.

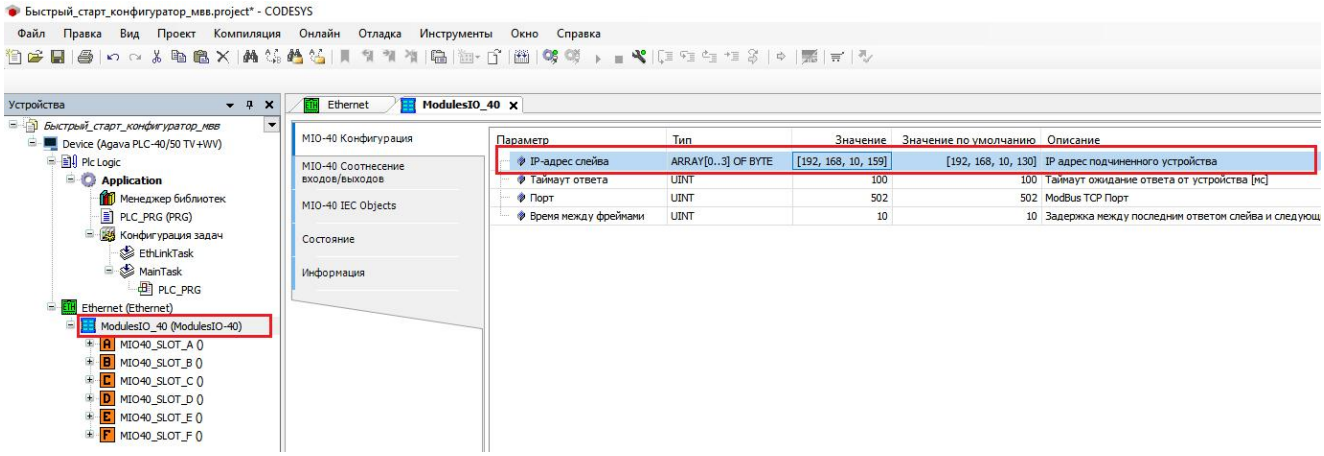


Рисунок 29. Настройка IP адреса устройства

Далее подключаем необходимые субмодули к слотам корзины МВВ-40, аналогично способу рассмотренному в разделе **5.1**

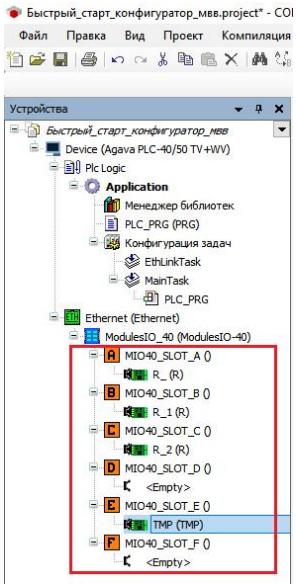


Рисунок 30. Подключение субмодулей к слотам корзины МВВ-40

Компилируем (F11), загружаем (Alt+F8) и выполняем проект (F5).

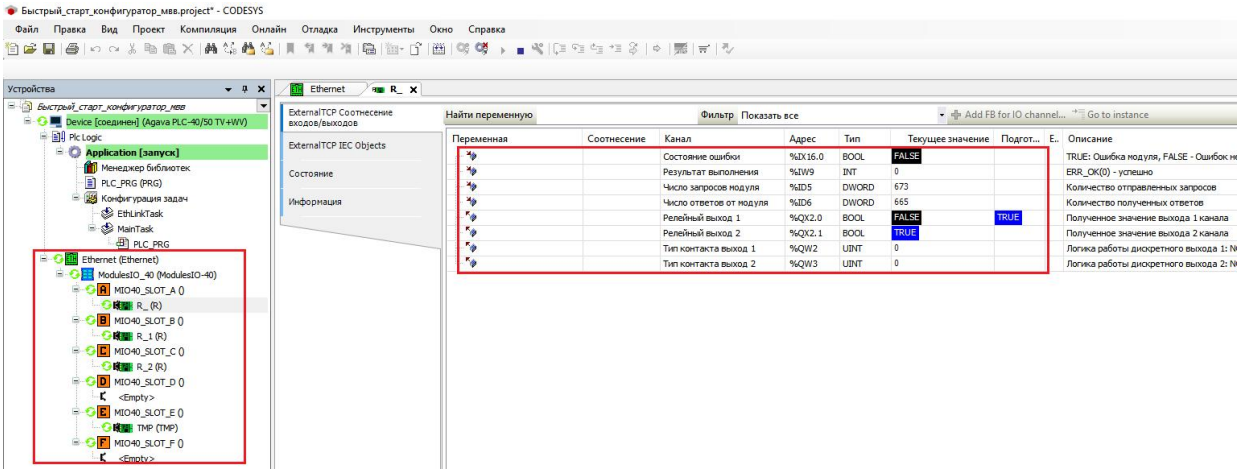


Рисунок 31. Выполнение проекта

Дополнительные модули расширения добавляются в проект аналогичными способом, кликом правой клавиши мыши по **Ethernet -> Добавить устройство.** Каждому новому устройству соответственно назначается свой IP адрес.

5.3. Проект с использованием субмодулей ПЛК-40

Рассмотрим пример создания проекта с использованием ПЛК-40, содержащего субмодули дискретных входов и релейных выходов. В качестве задания возьмём управление двумя насосами при помощи пускателей. Пусть нам необходимо включать и отключать два насоса с индикацией их текущего состояния. Для управления пускателями выберем субмодуль R, имеющий два релейных выхода, а для считывания обратной связи установим субмодуль с дискретными входами DI.

Проект рассматриваемого примера можно взять в SDK: Проекты\ПЛК-40\Быстрый старт\Быстрый старт.project

При открытии проектов из примеров SDK либо других сторонних проектов, могут возникнуть ошибки связанные с отсутствием некоторых версии библиотек, если это библиотеки из набора SDK Agava, то такую библиотеку нужно обновить через менеджер библиотек, нажав ПКМ по нужной библиотеке выбрать пункт меню «свойство», в открывшемся окне указать актуальную версии библиотеки, нажать «ок».

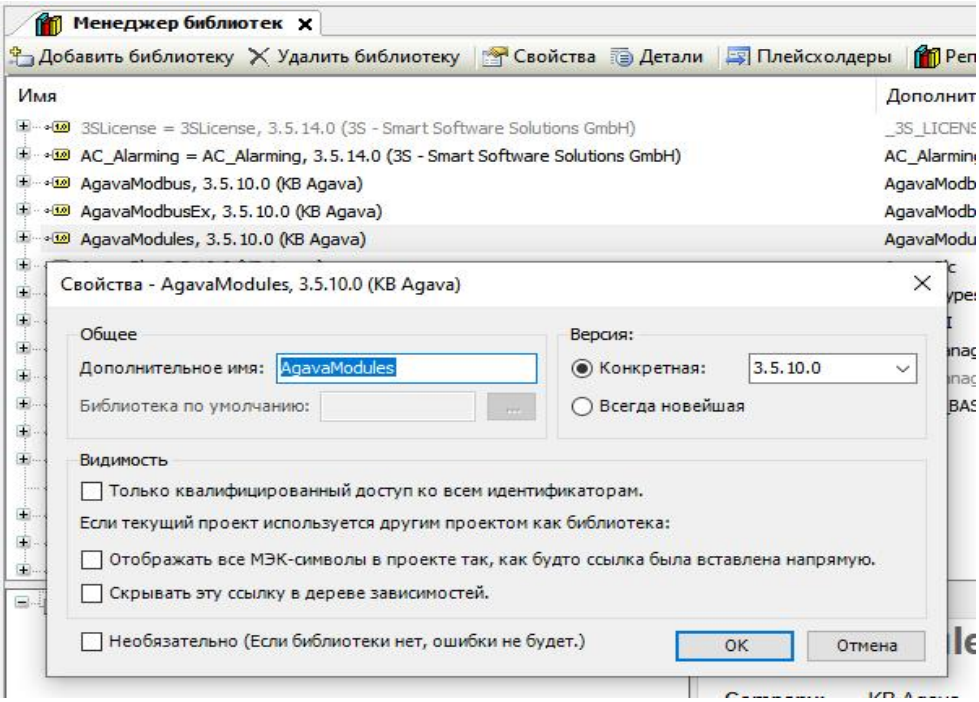


Рисунок 32. обновление библиотеки

Если в проекте отсутствует библиотека компании 3S, то её также необходимо установить используя менеджер библиотек, нажав кнопку «Загрузка отсутствующих библиотек». Данную кнопку необходимо нажимать до тех пор, пока она не исчезнет из менеджера библиотек, так как после загрузки нужных библиотек могут появляться ссылки на другие отсутствующие библиотеки. Если данная кнопка отсутствует в менеджере библиотек, то все необходимы библиотеки установленны в среду разработки CODESYS.

Создадим стандартный проект, как указано в п. 4.2 и подключим его к контроллеру. Добавим в проект библиотеку “AgavaModules” при помощи менеджера библиотек.

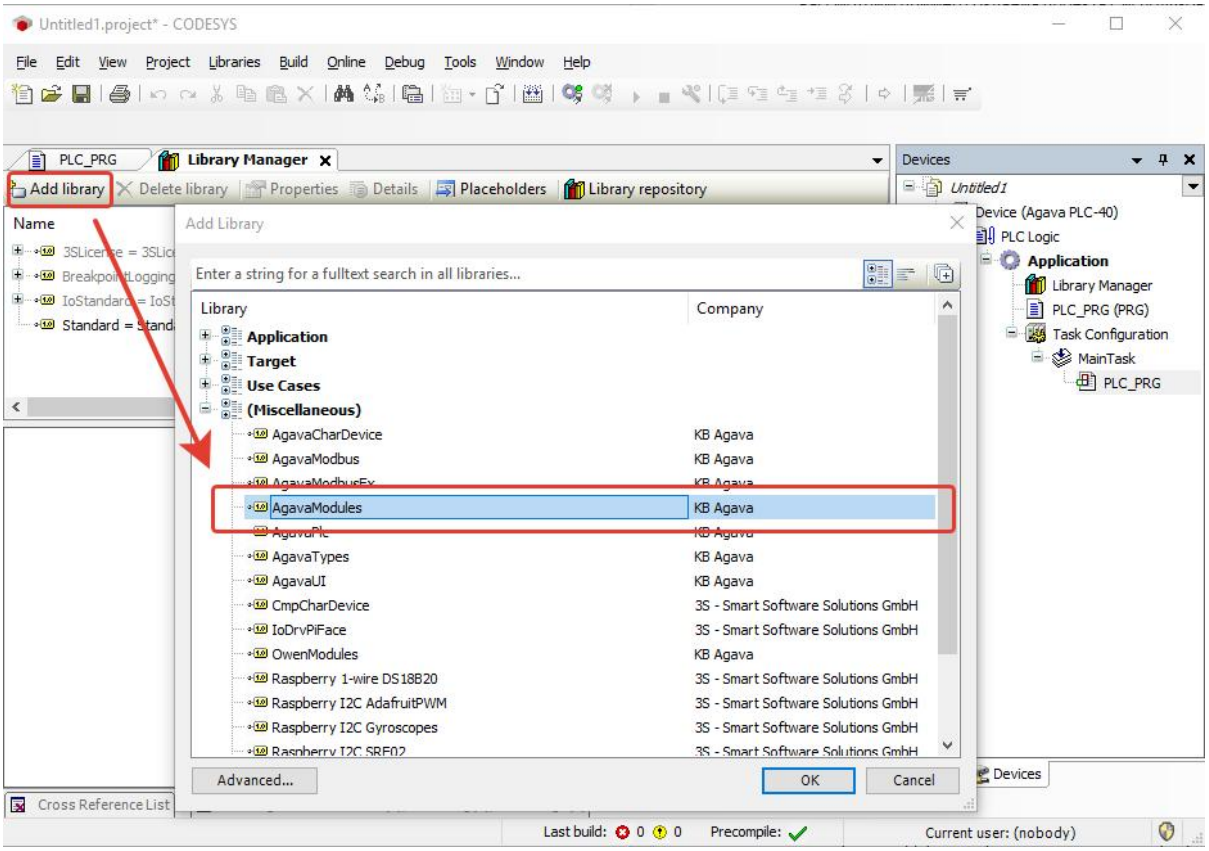


Рисунок 32-1. Добавление библиотеки

Выберем в дереве проекта элемент с программой PLC_PRG. Добавим в область объявления следующий код:

```
program PLC_PRG
var
    DI: TIntIoModuleSync_DI;    // Экземпляр фб модуля дискретных входов.
    MR: TIntIoModuleSync_R;     // Экземпляр фб модуля релейных выходов.
end_var
```

В область тела программы добавим код:

```
// Таблица соответствия слотов и адресов модулей ПЛК-40:
// Адрес: 1,   2,   3,   4,   5
// Слот: 'D', 'A', 'E', 'B', 'F'

// /-----------------------------------------------------------\
// |                                                           |
// |  +--------------------+        +--------------------+  |
// |  |      Слот 'A'      |        |      Слот 'D'      |  |
// |  +--------------------+        +--------------------+  |
// |                                                           |
// |  +--------------------+        +--------------------+  |
// |  |      Слот 'B'      |        |      Слот 'E'      |  |
// |  +--------------------+        +--------------------+  |
// |                                                           |
// |          Слот 'C'                                         |
// |  +--------+      +-----+        +--------------------+  |
// |  |        |      |     |        |      Слот 'F'      |  |
// |  +--------+      +-----+        +--------------------+  |
// |       Модуль питания                                      |
// \-----------------------------------------------------------/
//                      Задняя крышка ПЛК-40

// Выполняем обмен с модулем дискретных входов.
DI( slot := SLOT_A ); // Слот 'A'

// Выполняем обмен с модулем релейных выходов.
MR( slot := SLOT_B ); // Слот 'B'
```

Добавляем визуализацию в дерево проекта. Ставим галочку “использовать строки Unicode”.

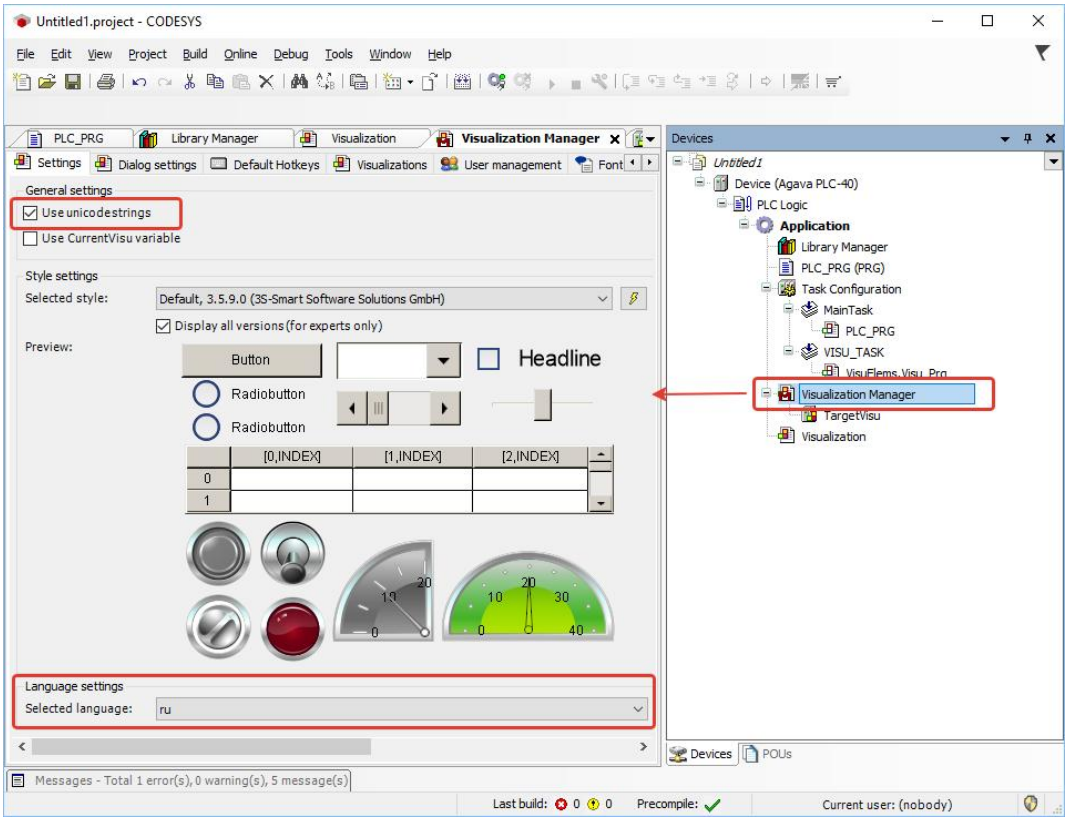


Рисунок 32-2. Установка поддержки unicode

Добавляем два тумблера и две лампы в визуализацию.

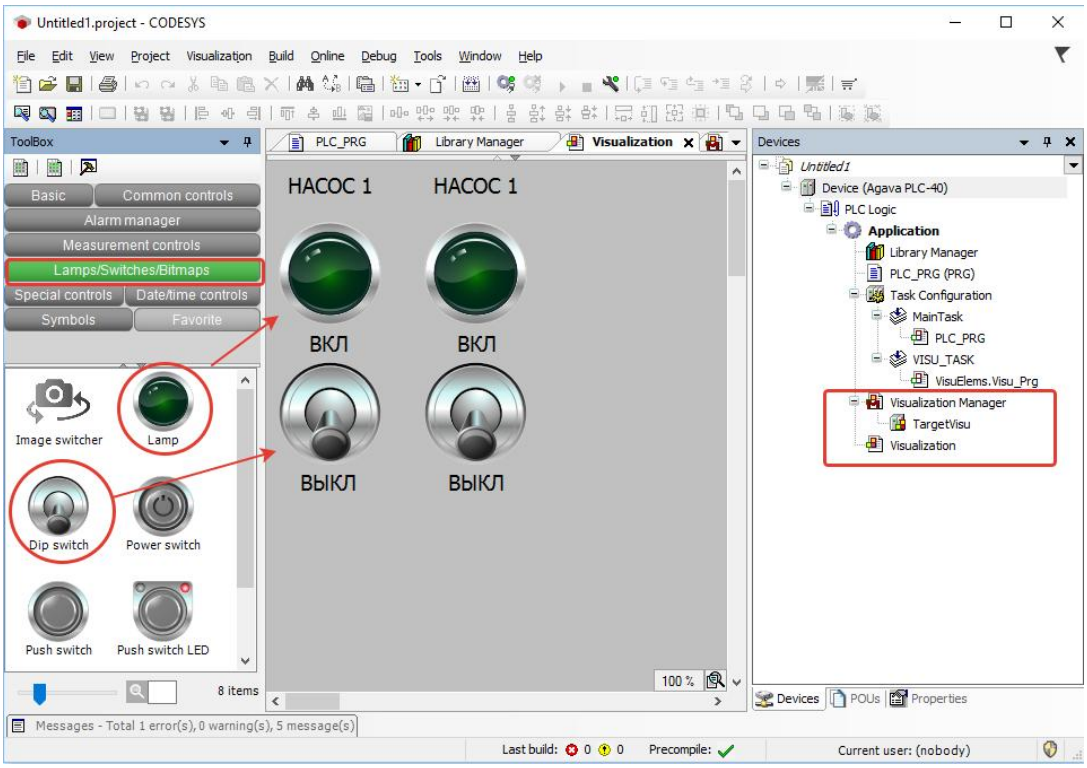


Рисунок 32-3. Добавление компонентов визуализации

Связываем свойства Variable компонентов с переменными функциональных блоков DI и MR.

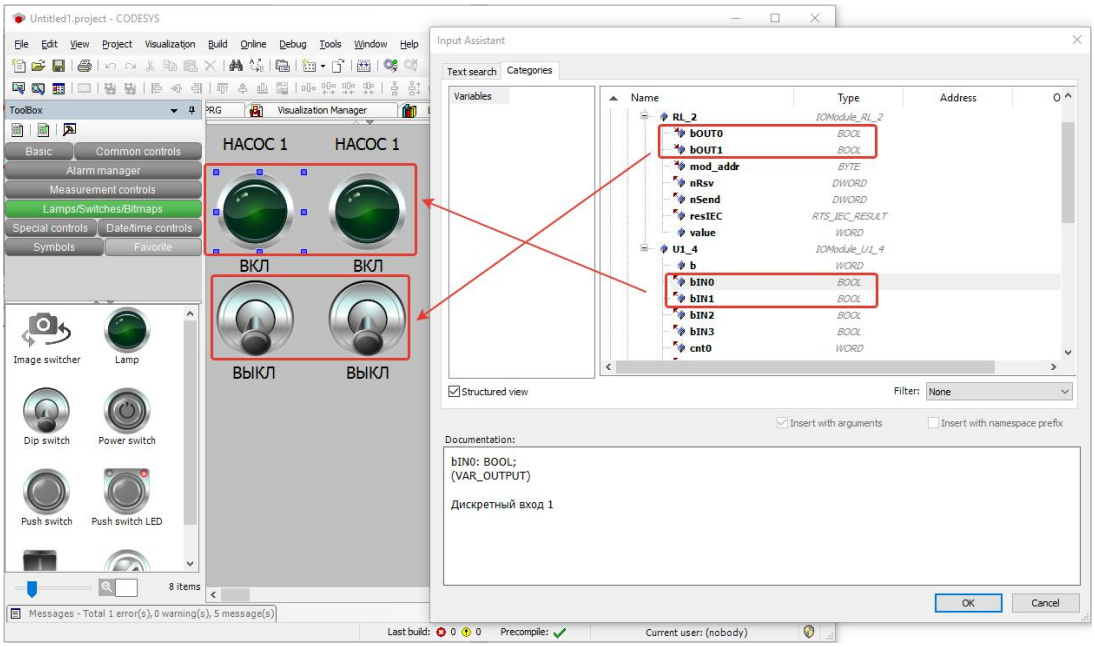


Рисунок 32-4. Связывание компонентов с переменными

Компилируем (F11), загружаем (Alt+F8) и выполняем проект (F5). Для проверки работы релейных выходов можно соединить их с соответствующими дискретными входами.

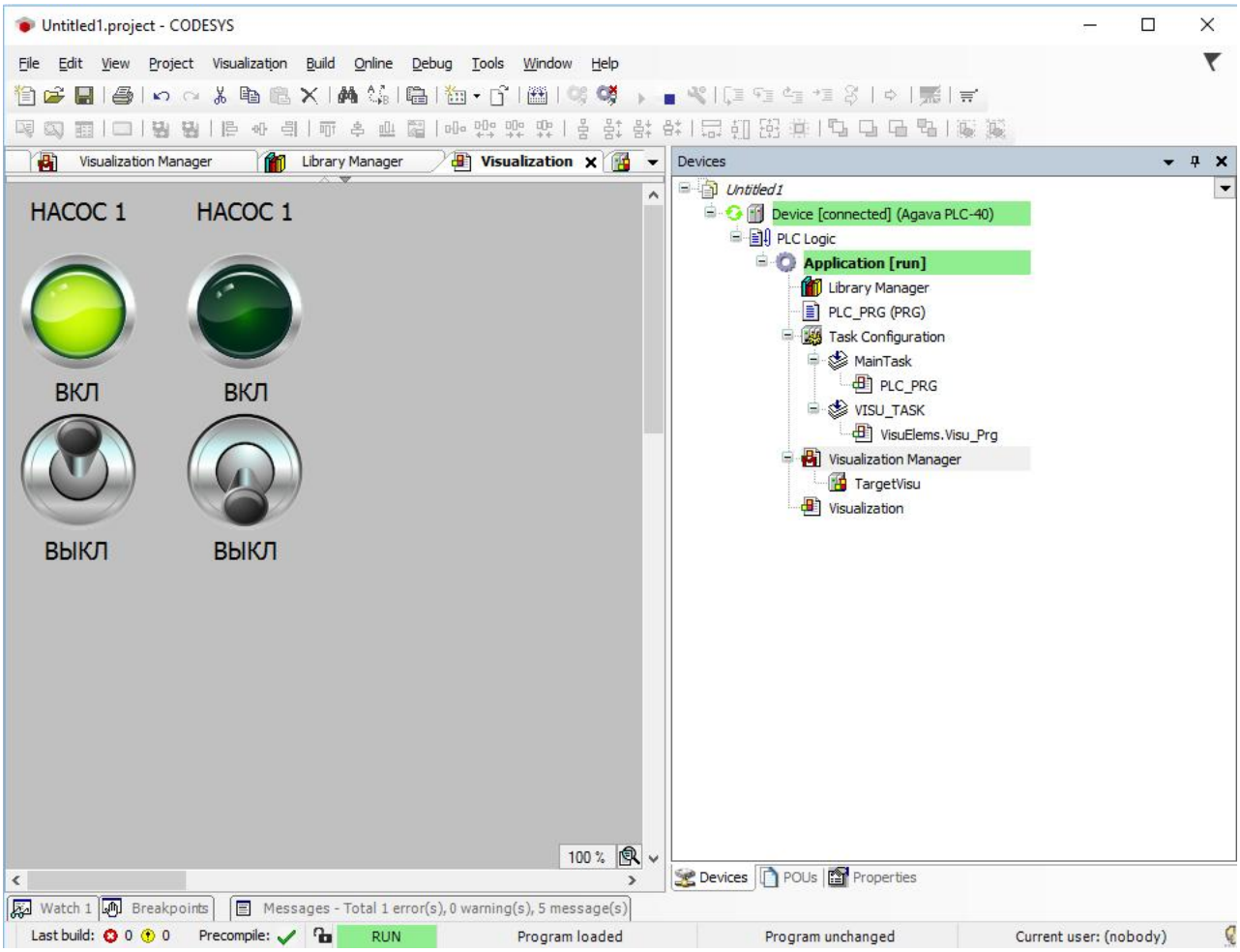


Рисунок 32-5. Отладка проекта в Codesys

5.4. Проект с использованием МВВ-40.3

5.4.1. Настройка сетевого взаимодействия

Перед установлением связи с МВВ-40.3 необходимо выполнить согласование сетей ПЛК-40 и МВВ-40.

По умолчанию ПЛК-40 настроен на получение сетевого адреса от DHCP-сервера, поэтому прямое соединение Ethernet кабелем ПЛК и МВВ работать не будет.

МВВ-40 по умолчанию имеет IP-адрес 192.168.10.130. Для установления соединения ПЛК и МВВ необходимо соблюдение минимум **одного из условий**:

1. Наличие на сетевом интерфейсе ПЛК адреса из сети 192.168.10.x
2. Наличие в локальной сети маршрутизатора, обеспечивающего доступ в сеть 192.168.10.x

При несоблюдении указанного условия соединение ПЛК и МВВ не будет установлено.

Самым простым способом настройки связи будет установка статического IP-адреса в настройках сетевого интерфейса ПЛК-40. IP-адрес ПЛК должен принадлежать сети 192.168.10.x, например, 192.168.10.100.

Просмотреть IP-адрес и другую сетевую конфигурацию для всех интерфейсов Ethernet можно из консоли, набрав команду:

```
ifconfig
```

```
root@agava6432_40:~# ifconfig
eth0      Link encap:Ethernet  HWaddr A8:1B:6A:48:71:93
          inet addr:192.168.10.215  Bcast:192.168.10.255  Mask:255.255.255.0
          inet6 addr: fe80::aa1b:6aff:fe48:7193%132688/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:865 errors:0 dropped:16 overruns:0 frame:0
          TX packets:98 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:75565 (73.7 KiB)  TX bytes:13702 (13.3 KiB)
          Interrupt:173

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1%132688/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:6 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:540 (540.0 B)  TX bytes:540 (540.0 B)

usb0      Link encap:Ethernet  HWaddr 46:10:3A:B3:AF:D9
          inet addr:192.168.7.1  Bcast:192.168.7.3  Mask:255.255.255.252
```

```
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@agava6432_40:~#
```

Задать статический IP-адрес интерфеса eth0 можно используя системную утилиту ПЛК.

Также статический IP-адрес интерфеса eth0 можно задать в файле /etc/systemd/network/10-eth.network, например:

```
[Network]
DHCP=no
Address=192.168.10.100/24
Gateway=192.168.10.10
```

Для редактирования файла можно воспользоваться встроенным фаловым менеджером mc, либо установить подключение к ПЛК через USB интерфейс (адрес ПЛК - 192.168.7.1) по протоколу SFTP и изменить файл имеющимся в ОС текстовым редактором.

5.4.2. Создание и запуск проекта

Рассмотрим пример создания проекта с использованием МВВ-40.3, содержащего два субмодуля дискретных входов (DI) и один субмодуль релейных выходов (R).

Проект рассматриваемого примера можно взять в SDK: Проекты\ПЛК-40\Быстрый старт\Быстрый старт - MBB40.project

Создадим стандартный проект, как указано в разделе 4.3 настоящего руководства, и подключим его к контроллеру. Добавим в проект библиотеку “AgavaModules” и “AgavaTypes” при помощи менеджера библиотек.

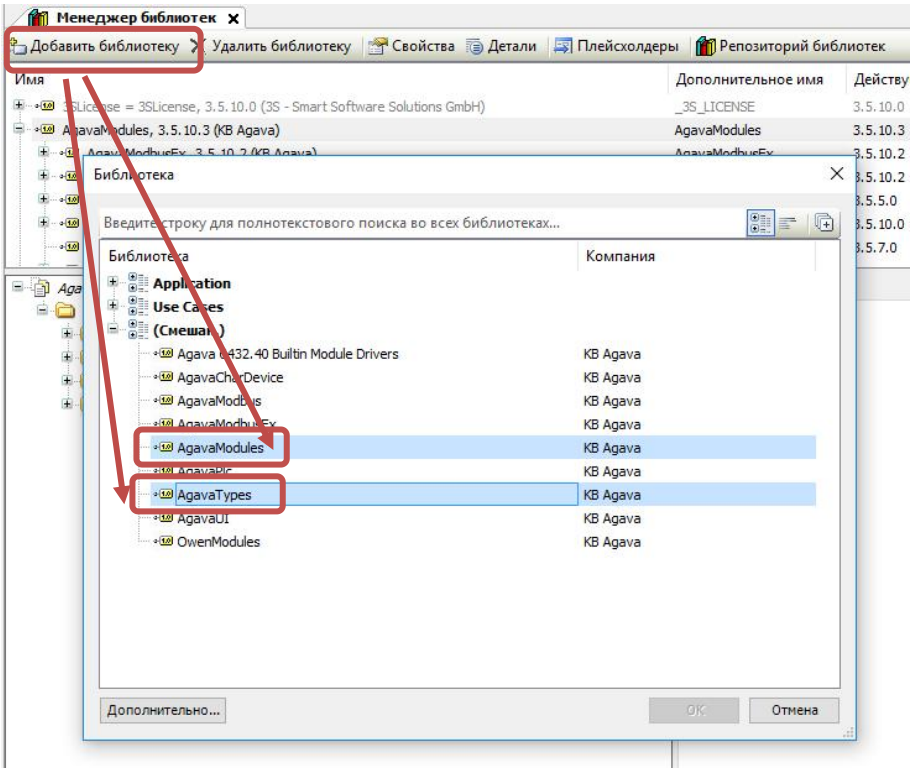


Рисунок 32-6. Добавление библиотек

Выберем в дереве проекта элемент с программой PLC_PRG. Добавим в область объявления следующий код:

```
program PLC_PRG
var
    MDI_1: TIntIoModuleSync_DI;         // Экземпляр фб модуля 1 дискретных входов.
    MDI_2: TIntIoModuleSync_DI;         // Экземпляр фб модуля 2 дискретных входов.
    MR_1: TIntIoModuleSync_R;           // Экземпляр фб модуля 1 релейных выходов.
    bErr: bool;
    slaveId: byte;
    port: uint;
    cntSend, cntRecv: dword;
    ioparam: udint;
    start, stop: ulint;
    timeout: time;
    ipAddress: string(16);
    result: AgavaTypes.RTS_IEC_RESULT;
    socket: TSocket;
    MDI1cnt2: WORD;
    MDI1cnt3: WORD;
    MDI2cnt2: WORD;
    MDI2cnt3: WORD;
end_var
```

В область тела программы добавим код:

```
//Настройка параметров соединения Ethernet
```

```
     timeout := t#50ms;                                  // Таймаут ожидания ответа.
     ipAddress := '192.168.10.130';                       // IP адрес MBB-40(адрес по умолчанию настроенный
в MBB).
     port := 502;                                        // Порт сервера (502 для Modbus TCP).
     result := socket.Create();                          // Создаём сокет.
          if result = 0 then
                result := socket.Connect( ipAddress, port );    // Подключаемся к MBB-40.
                ioparam := 1;
                result    :=    AgavaTypes.SysSockIoctl(    socket.Handle,    AgavaTypes.SOCKET_FIONBIO,
adr( ioparam ));
          end_if
     slaveId := 1;                                       // Адрес подчинённого устройства MBB-40.
     AgavaTypes.SysTimeRtcHighResGet( start );           // Метка начала запроса.

(*Добавление субмодуля дискретных выходов типа "реле" R модуль 1*)
 // Параметры связи.
      MR_1.linktype := EnLinkType.ltSocket;              // тип линии
      MR_1.handle := socket.Handle;                      // дескриптор соединения
      MR_1.devid := slaveId;                             // адрес прибора в сети modbus
      MR_1.modno :=1;                                    // порядковый номер субмодуля R в группе(1 до 6)
      MR_1.timeout := timeout;                           // интервал ожидания ответа
      // Выполняем синхронный запрос.
      MR_1( xEnable := true, xError => bErr, resIEC => result, nSend => cntSend, nRsv => cntRecv );

(*Добавление субмодуля дискретных входов типа "Сухой контакт" DI - модуль 1*)
  // Параметры связи.
       MDI_1.linktype := EnLinkType.ltSocket;            // тип линии
       MDI_1.handle := socket.Handle;                    // дескриптор соединения
       MDI_1.devid := slaveId;                           // адрес прибора в сети modbus
       MDI_1.modno := 1;                                 // порядковый номер субмодуля DI в группе (1 до 6)
       MDI_1.timeout := timeout;                         // интервал ожидания ответа
 // Значения счётчиков.
       MDI1cnt2 := MDI_1.cnt2;
       MDI1cnt3 := MDI_1.cnt3;
// Выполняем синхронный запрос.
       MDI_1( xEnable := true, xError => bErr, resIEC => result, nSend => cntSend, nRsv => cntRecv );

          (*Добавление субмодуля дискретных входов типа "Сухой контакт" DI - модуль 2*)
 // Параметры связи.
        MDI_2.linktype := EnLinkType.ltSocket;           // тип линии
        MDI_2.handle := socket.Handle;                   // дескриптор соединения
        MDI_2.devid := slaveId;                          // адрес прибора в сети modbus
        MDI_2.modno := 2;                                // порядковый номер субмодуля DI в группе(1 до 6)
        MDI_2.timeout := timeout;                        // интервал ожидания ответа
 // Значения счётчиков.
        MDI2cnt2 := MDI_2.cnt2;
        MDI2cnt3 := MDI_2.cnt3;
 // Выполняем синхронный запрос.
        MDI_2( xEnable := true, xError => bErr, resIEC => result, nSend => cntSend, nRsv => cntRecv );

 // Метка ответа или окончания времени ожидания.
       AgavaTypes.SysTimeRtcHighResGet( stop );
```

Добавляем визуализацию в дерево проекта. Ставим галочку “использовать строки Unicode”.

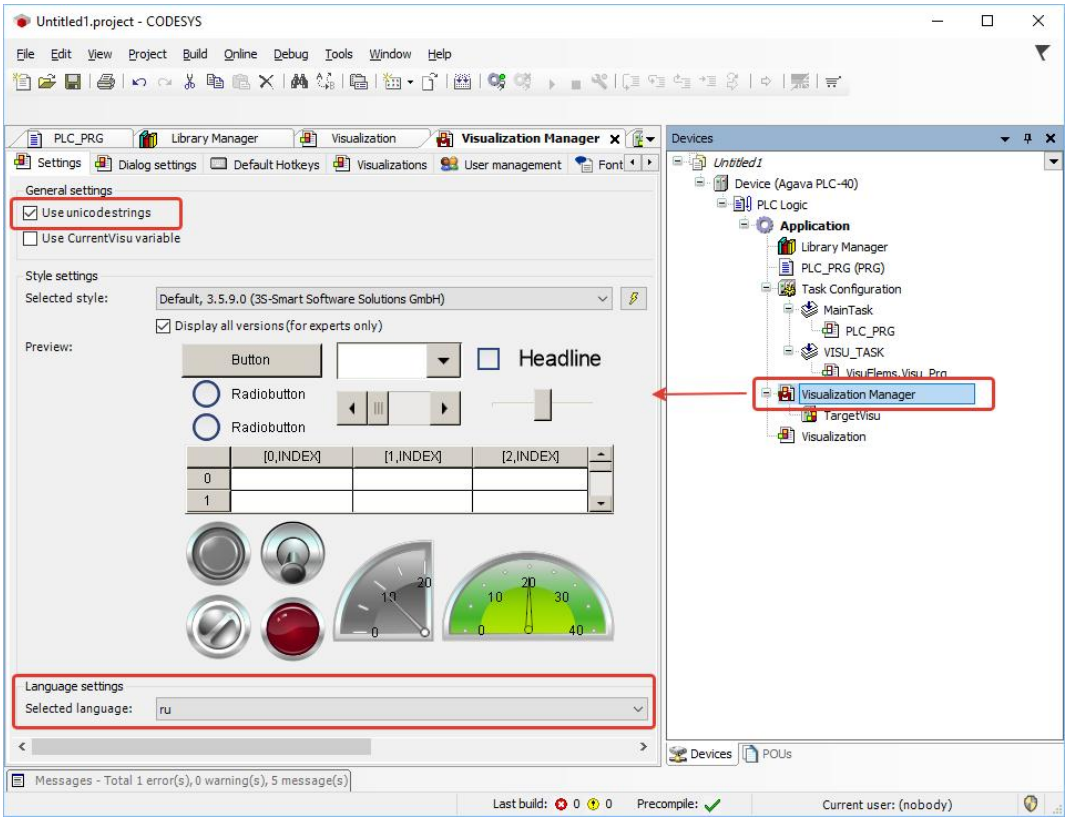


Рисунок 32-7. Установка поддержки unicode

Добавляем тумблеры и лампы в визуализацию.

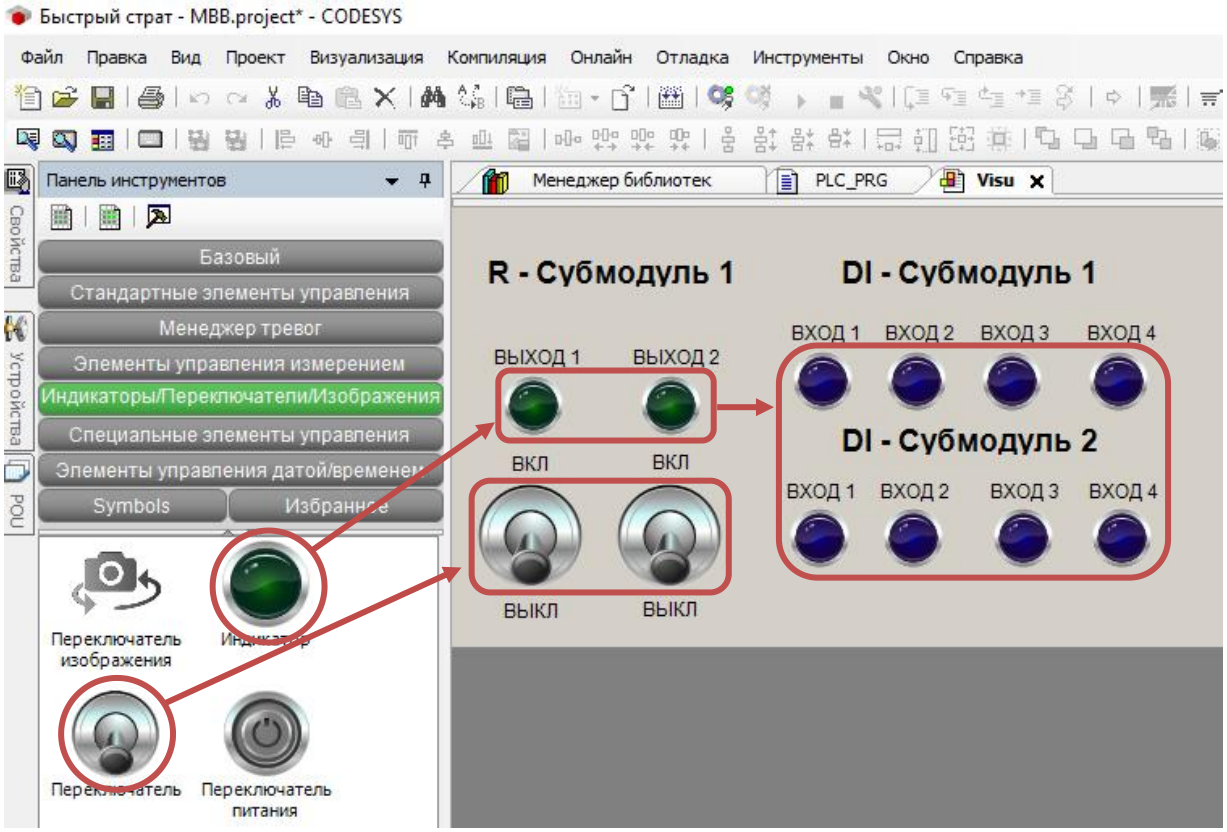


Рисунок 32-8. Добавление компонентов визуализации

Связываем свойства «Переменная» компонентов с переменными функциональных блоков DI и MR.

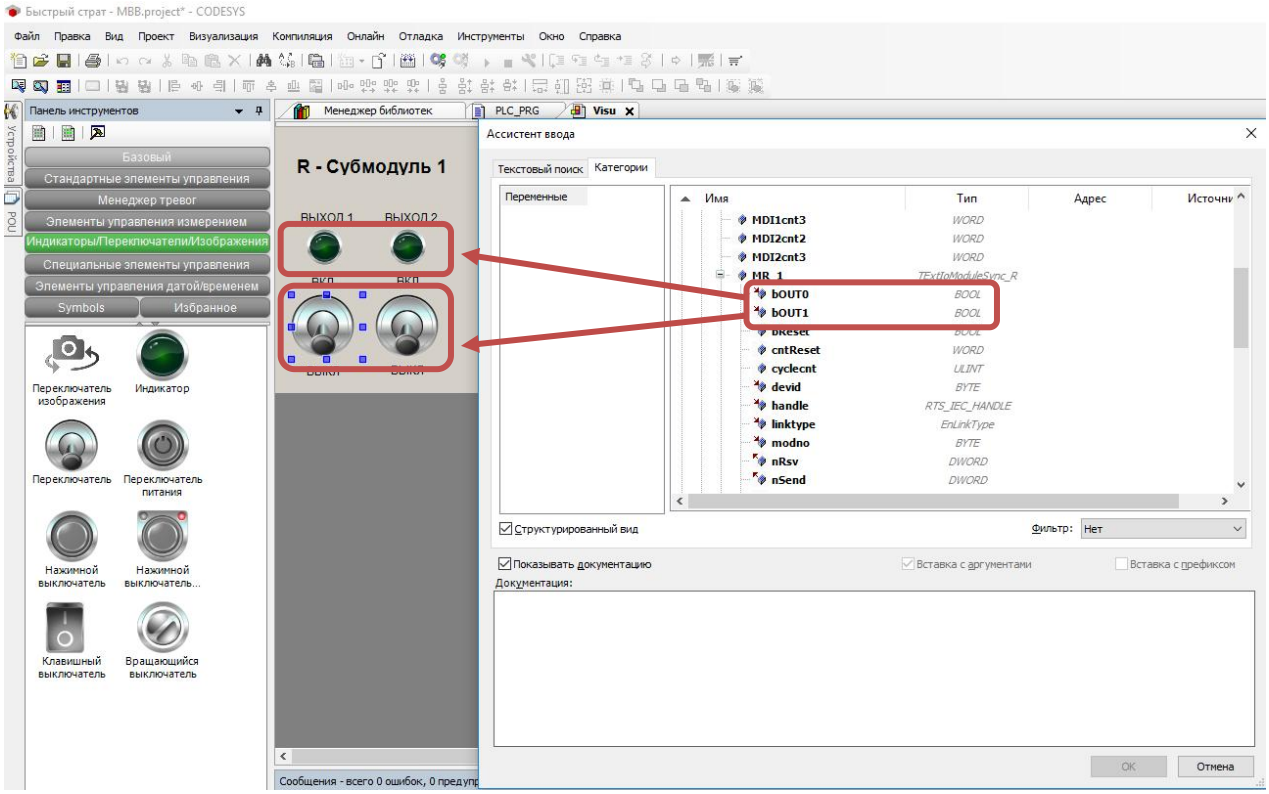


Рисунок 32-9. Связывание компонентов с переменными R модуля 1

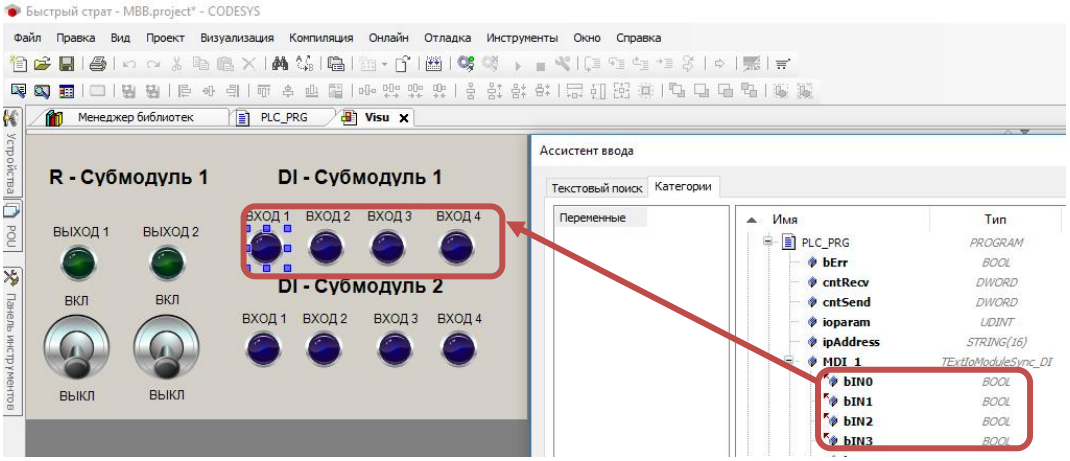


Рисунок 32-10. Связывание компонентов с переменными DI модуля 1

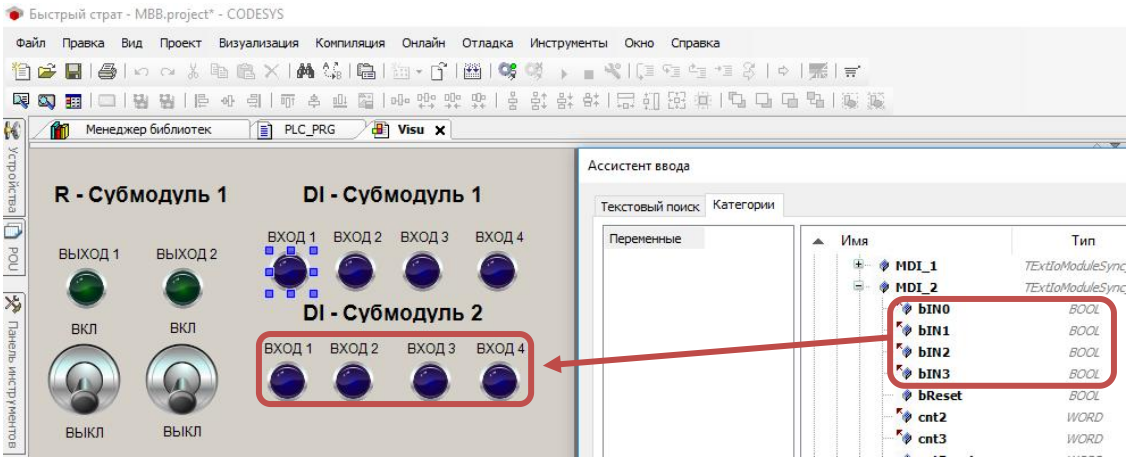


Рисунок 32-11. Связывание компонентов с переменными DI модуля 2

В результате связывания в свойстве «перерменная», визуальных компонентов дожно получитсья следующие:

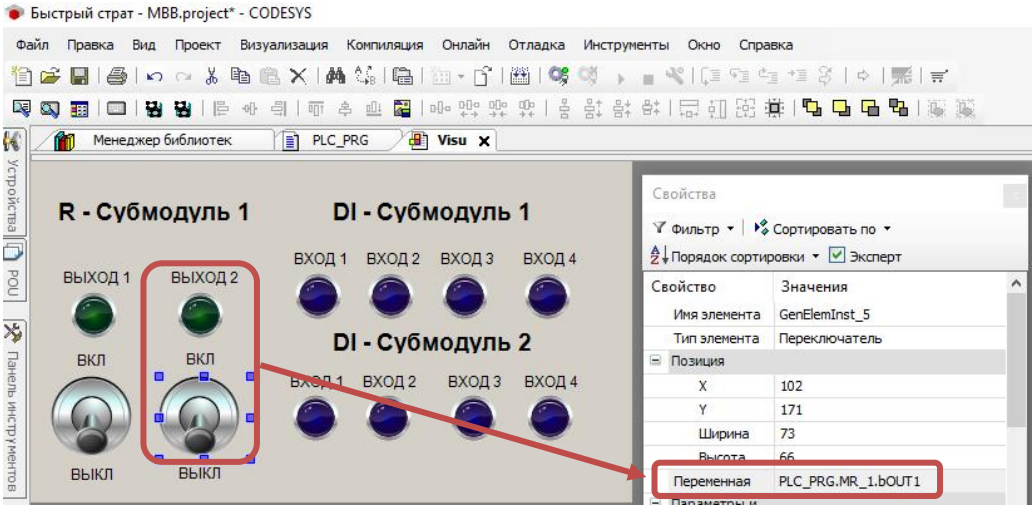


Рисунок 32-12. Связывание компонентов с переменными

Компилируем (F11), загружаем (Alt+F8) и выполняем проект (F5). Для проверки работы переключить тумлер дискретного выхода в положение ВКЛ.

Для проверки работы дискретных входов подключить кнопочные переключатели согласно руководству по эксплуатации ПЛК-40.

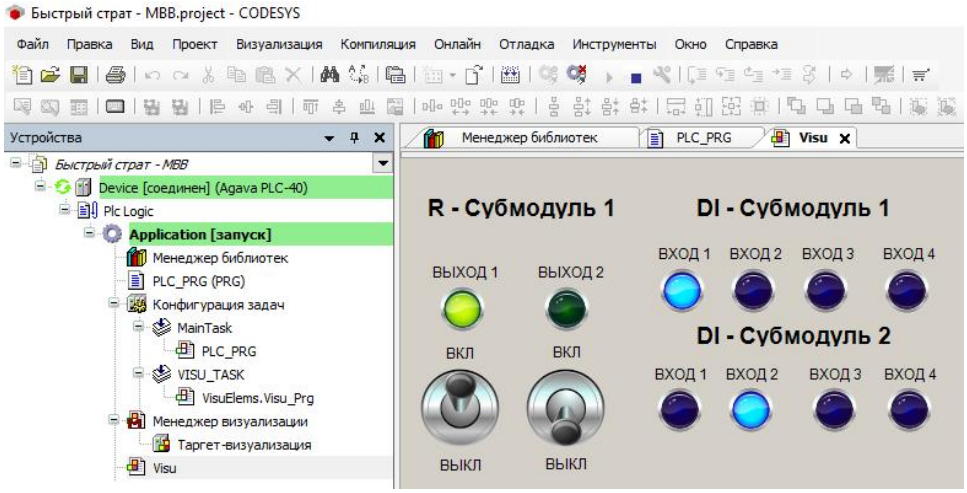


Рисунок 32-13. Отладка проекта в Codesys

5.5. Работа с энергонезависимыми переменными RETAIN и PERSISTENT

Рассмотрим пример использованием энергонезависимой памяти для сохранения переменных в случае отключения питания ПЛК.

Создадим стандартный проект, как указано в п. 4.3. При создании в дерево проекта по умолчанию встраивается устройство **RetainStorage**, необходимое для работы с энергонезависимыми переменными.

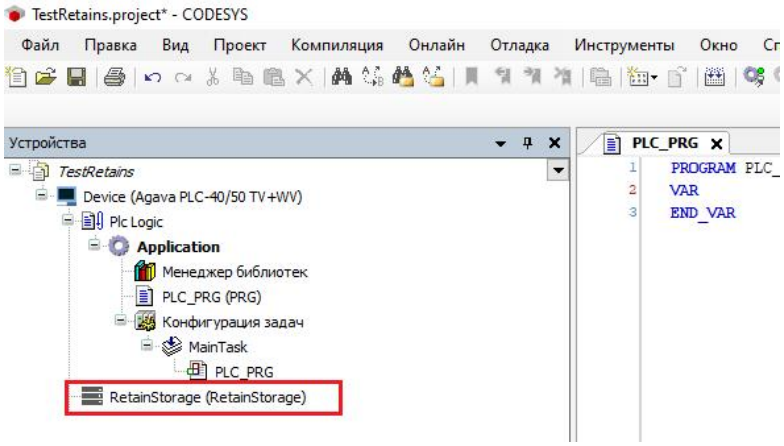


Рисунок 33. RetainStorage - устройство для работы с энергонезависимыми переменными.

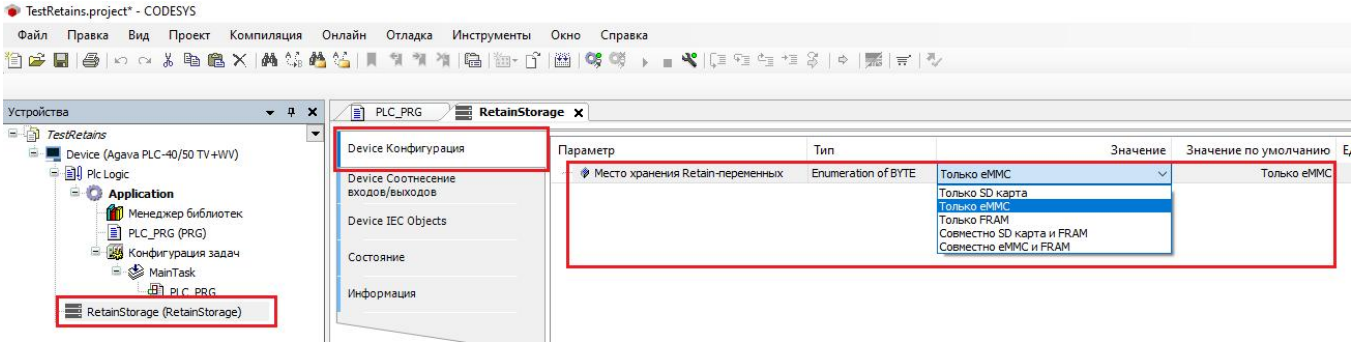


Рисунок 34. RetainStorage конфигурация устройства.

Двойным кликом левой клавиши мыши откроем настройки устройства и перейдём на вкладку «**Конфигурация**». Далее укажем место хранения **Retain** переменных, доступно несколько вариантов сохранения переменных.

Только SD карта

Переменные будут храниться на SD накопителе. Накопитель должен иметь один раздел и должен быть отформатирован в файловую систему **Ext4 с помощью системной утилиты ПЛК.** Область памяти в среде исполнения CODESYS: RETAIN.

Только eMMC

Переменные будут храниться внутри файловой системы ПЛК, на eMMC накопителе. Область памяти в среде исполнения CODESYS: RETAIN.

Только FRAM

Переменные будут храниться в памяти FRAM. Максимальный объем накопителя FRAM 8 кб. Область памяти в среде исполнения CODESYS: PERSISTENT.

Совместно SD карта и FRAM

Переменные будут храниться совместно на SD накопителе и в памяти FRAM. Максимальный объем накопителя FRAM 8 кб. Области памяти в среде исполнения CODESYS: RETAIN, PERSISTENT.

Совместно eMMC и FRAM

Переменные будут храниться совместно в файловой системе ПЛК и в памяти FRAM. Максимальный объем накопителя FRAM 8 кб. Области памяти в среде исполнения CODESYS: RETAIN, PERSISTENT.

Если в проекте предусматривается сохранение переменных, которым свойственно редкое изменение, например уставки или переменные конфигурации, то в таком случае разумно использовать место хранение **Только SD карта** или **Только eMMC.** В других случаях если требуется сохранять более динамичные переменные, которые часто подвержены изменению, то нужно использовать место хранение **Только FRAM,** если в проекте есть оба типа переменных, тогда нужно выбирать совместное использование накопителей. В качестве примера в конфигурации **RetainStorage** выберем место хранения **Только SD карта.**

Для сохранения переменной в области энергонезависимой памяти RETAIN, добавим в проект объект «**Список глобальных переменных**», для этого кликнем правой клавишей мыши по «**Application**» в открывшемся контекстном меню выберем пункт «**Добавление объекта**» далее «**Список глобальных переменных**» и нажмём кнопку «**Добавить**».

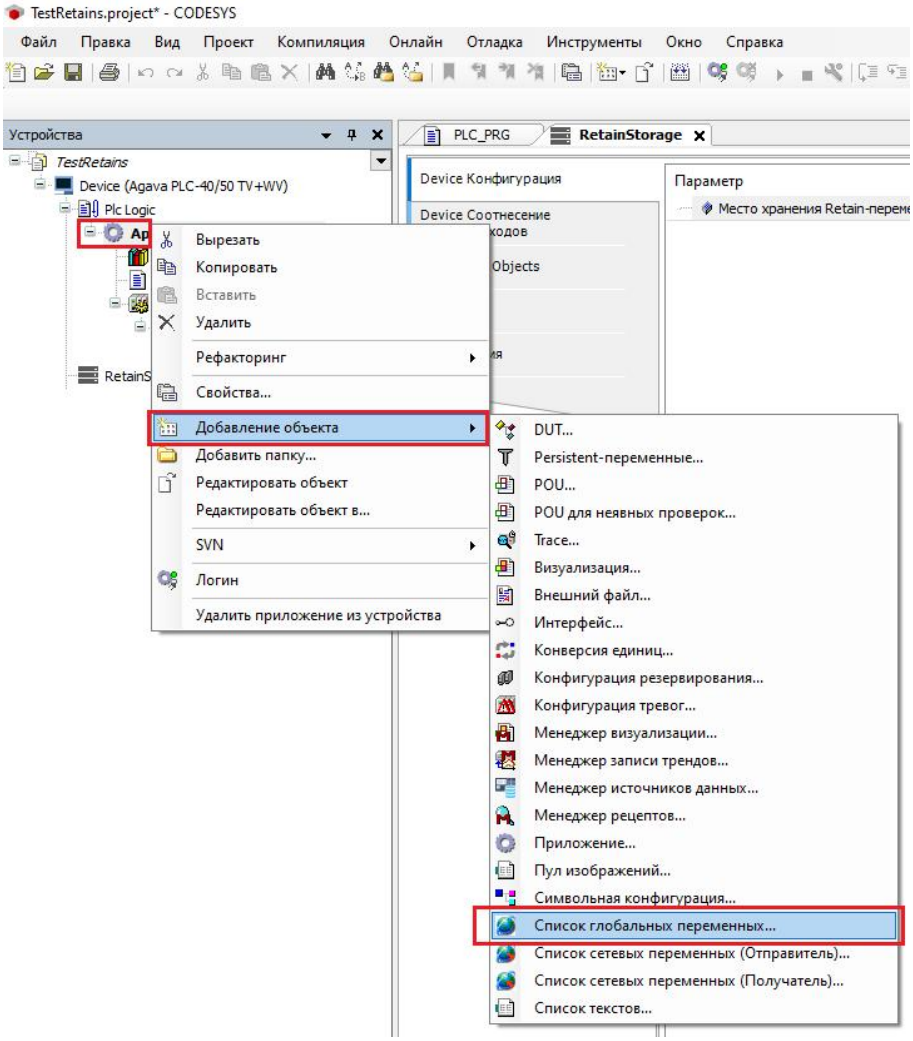


Рисунок 35. Добавление списка глобальных переменных.

В список глобальных переменных добавим новую область **VAR_GLOBAL RETAIN** и в ней объявим тестовый массив значений.

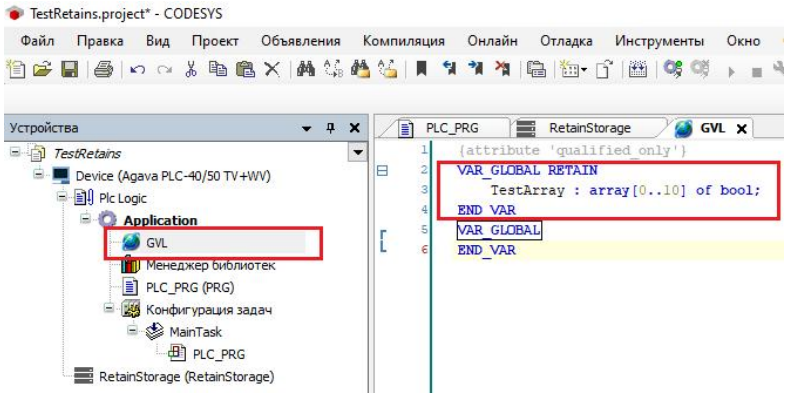


Рисунок 36. Добавление энергонезависимых переменных.

В основой программе **PLC_PRG** нашем несколько строк кода для проверки.

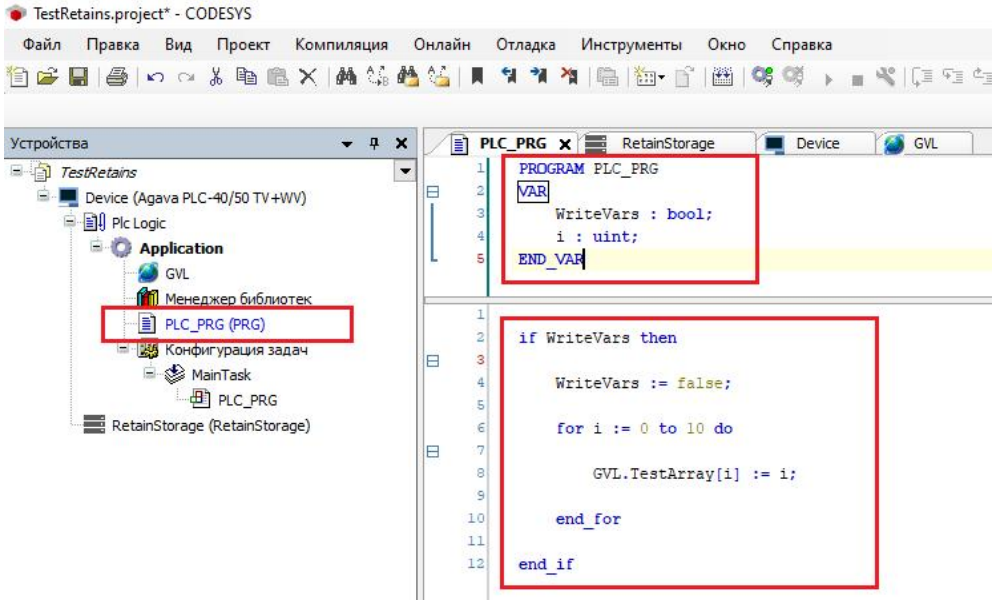


Рисунок 37. Проверка записи переменных.

Далее с компилируем (F11), загружаем (Alt+F8) и выполняем проект (F5).

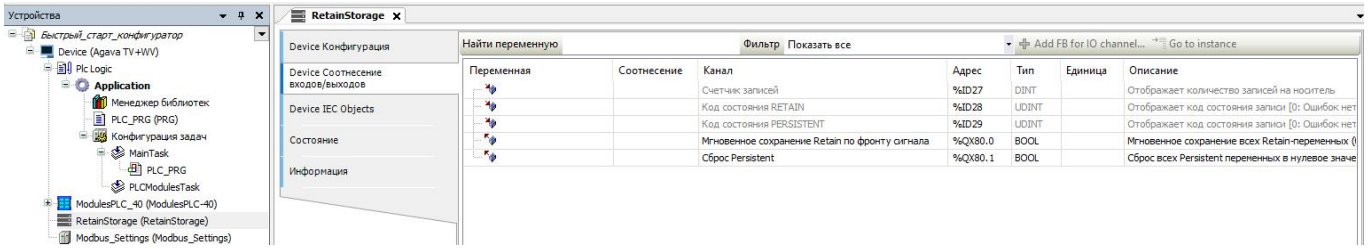


Рисунок 37-1. Проверка записи переменных.

В основной программе включим флаг **WriteVars,** произойдёт запись на SD накопитель, в момент записи зелёное кольцо на против **RetainStorage** примет вид «серого треугольника», счётчик записей увеличится. В тестовый массив запишутся значения итератора цикла FOR.

Выключим питание контроллера, затем через несколько секунд подадим питание вновь. После загрузки контроллера убедимся что тесовый массив содержит ранее записанное значение.

Для использования **PERSISTENT** области и **FRAM** накопителя в проект необходимо добавить «**Persistent - переменные**». Для этого правой клавишей мыши кликнем по «**Application**», выберем пункт контекстного меню «**Добавление объекта**», затем пункт «**Persistent - переменные**».

В области **VAR_GLOBAL PERSISTENT RETAIN** и в ней объявим тестовый массив значений.

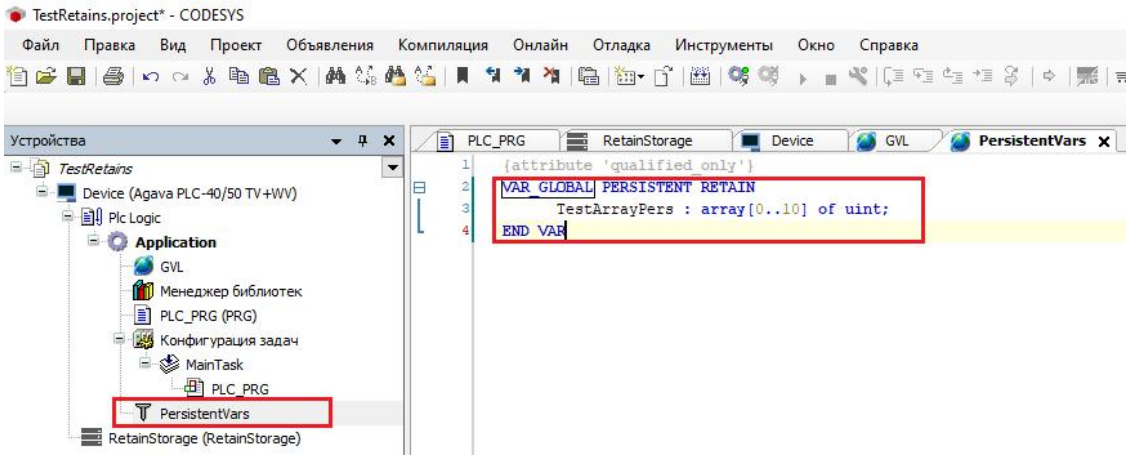


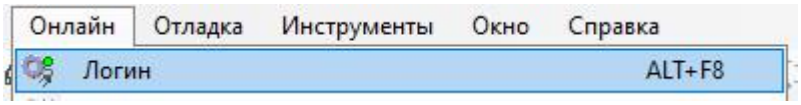
Рисунок 38. Добавление энергонезависимых переменных PersistentVars.

В **RetainStorage** место хранения изменим на **Совместно SD карта и FRAM.** Далее с компилируем (F11), загружаем (Alt+F8) и выполняем проект (F5).

В тестовом массиве **TestArrayPers** изменяем значения , затем выключим питание контроллера, затем через несколько секунд подадим питание вновь. После загрузки контроллера убедимся что тесовый массив содержит ранее записанное значение.

FRAM накопитель работает только с областью памяти PERSISTENT RETAIN, если область не определена данные записать не получится.

Для полной очистки накопителя FRAM требуется подключиться к ПЛК



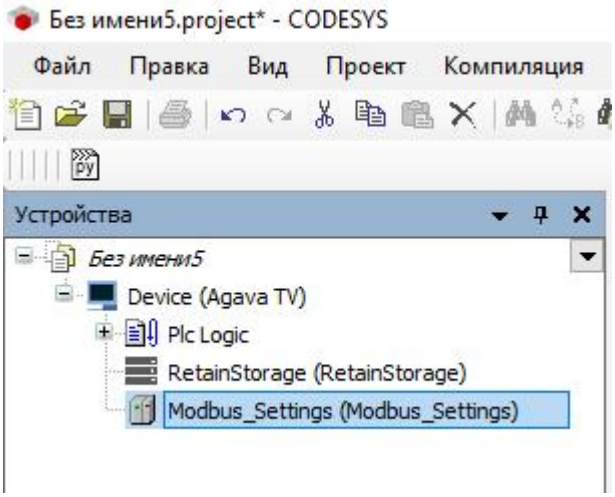
Открыть RetainStorage и установить параметр «Сброс Retain» в TRUE, после сброса установить параметр «Сброс Retain» в FALSE.

Переменная	Соотнесение	Канал	Адрес	Тип	Текущее значение
		Счетчик записей	%ID27	DINT	0
		Код состояния RETAIN	%ID28	UDINT	0
		Код состояния PERSISTENT	%ID29	UDINT	0
		Мгновенное сохранение Retain по фронту сигнала	%QX80.0	BOOL	FALSE
		Сброс Persistent	%QX80.1	BOOL	TRUE

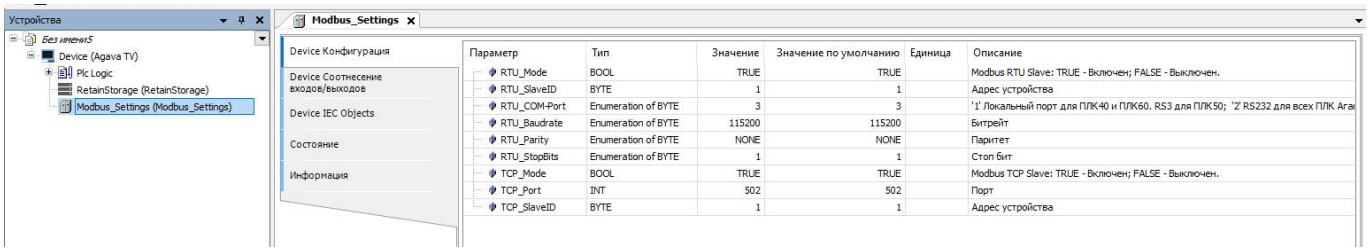
Максимальный объем доступный для записи у накопителя FRAM - 8 кб, при превышении допустимого размера накопителя, новые данные сохранены не будут, RetainStorage при этом будет возвращать при попытке сохранения код состояния 51.

5.6. Обмен по протоколу Modbus контроллеров Агава

Начиная с версии SDK 3.5.10.17 для всех ПЛК Агава доступен преднастроенный обмен по протоколу Modbus TCP и Modbus RTU. При создании проекта, либо при обновлении устройства(ПЛК) в старом проекте, в дереве появляется устройство «Modbus_Settings»



Открыв его(двойной клик ЛКМ) появляется окно настроек.



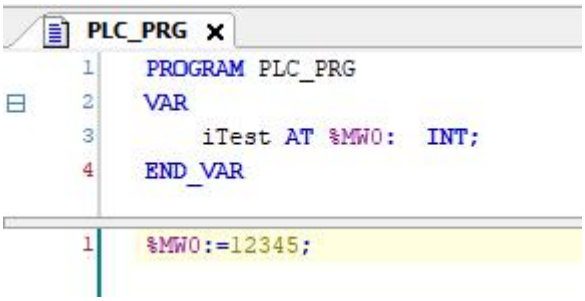
По желаю настройки можно изменить либо отключить обмен, если он не требуется либо планируется его реализация в коде проекта.

Обмен осуществляется с областью памяти HOLDING REGISTR в диапазоне адресов от %MW0 до %MW60000. Поддерживаются команды 03(0x03) Read Holding Registers, 06(0x06) Preset Single Register и 16(0x10) Preset Multiple Registers.

Распределение памяти:

- Один двойной регистр %MD0(любой 32 битный тип) содержит два регистра %MW0 и %MW1(любой 16 битный тип);
- Один регистр %MW0 содержит два байта %MB0 и %MB1;
- Один байт %MB0 содержит 8 бит %MX0.0-%MX0.7;
- Один байт %MB1 содержит 8 бит %MX1.0-%MX1.7;
- Один регистр %MW1 содержит два байта %MB2 и %MB3;
- Один байт %MB2 содержит 8 бит %MX2.0-%MX2.7;
- Один байт %MB3 содержит 8 бит %MX3.0-%MX3.7;

Адреса можно привязывать к переменным при объявлении либо использовать явно (без привязки к переменным) в коде программы:



При такой реализации любая переменная привязанная к %MW0 будет равна 12345.

Пример:

Допустим небходимо передать значение с плавающей точкой, для этого объявляем 32 битную переменную с типом REAL и привязываем ее к %MD0. При такой реализации мы можем передавать и принимать значение без дополнительных манипуляций для объединения двух 16 битных регистров в один 32 битный с использованием Union и/или указателей.

```
PROGRAM PLC_PRG
VAR
    iTest AT %MD0:  REAL;
END_VAR
```

Также переменные можно привязать к массиву и каждый элемент массива будет привязан к своему регистру iTest[0]=%MW0, iTest[1]=%MW1, iTest[2]=%MW2 и т.д.

```
VAR
    iTest AT %MW0: ARRAY[0..100] OF WORD;
END_VAR
```

6. Описание набора разработчика

6.1. Содержание архива

Набор разработчика (SDK - Software Development Kit) распространяется в виде архива и доступен для загрузки по адресу, указанному в п. 1.2 Полезные ссылки. В состав архива входят библиотеки и драйверы Codesys, файлы описания устройств, тестовые примеры, проекты и документация.

Папка «Документация\ПЛК-40»:
- паспорт (ПС);
- руководство по эксплуатации (РЭ);
- руководство программиста (РП).

Папка «Документация\ПЛК-50»:
- паспорт (ПС);
- руководство по эксплуатации (РЭ);
- руководство программиста (РП).

Папка «Документация\ПЛК-60»:
- паспорт (ПС);
- руководство по эксплуатации (РЭ);
- руководство программиста (РП).

Папка «Документация\МВВ-40»:
- паспорт;
- руководство по эксплуатации.

Папка «Примеры» содержит тестовые проекты для демонстрации работы с библиотеками Codesys. Папка «Проекты» содержит различные демонстрационные проекты.

Папка «Библиотеки» содержит установочный пакет Codesys. В нём находятся библиотеки и файлы описания устройств.

6.2. Порядок установки SDK

Перед открытием проектов необходимо:

1) Установить пакет «AgavaLibraries.x.x.x.x.package» (x.x.x.x – это номер версии), входящий в Agava SDK. Для этого нужно открыть менеджер пакетов в меню Tools, нажать кнопку Install и выбрать файл пакета в диалоговом окне. Пакет содержит в себе набор библиотек, драйверов и описаний устройств.
2) Установить драйвер RNDIS из Agava SDK с использованием файла «AgavaPLC. USB driver setup.exe». Этот драйвер, поддержка которого встроена в ОС Windows, создаёт виртуальную сетевую карту, доступную в Диспетчере устройств. На базе этой сетевой карты Windows создаёт сетевое подключение, которое работает через usb.

7. Описание библиотек Codesys

7.1. Библиотека AgavaTypes

Библиотека версии 3.5.10.0 содержит набор базовых и вспомогательных типов, используемых в других библиотеках. Типы, использующие динамическую память, требуют её включение в свойствах Application проекта.

7.1.1. Функциональный блок TBitmap

Точечное изображение.

Пространство имён: AgavaTypes.

7.1.1.1. Определение

```
{attribute 'enable_dynamic_creation'}
function_block TBitmap extends TImage
```

7.1.1.2. Свойства

Название	Тип	Описание
PixelFormat	EnPixelFormat	Возвращает формат изображения
Width	int	Возвращает ширину изображения в пикселях
Height	int	Возвращает высоту изображения в пикселях
Size	TSize	Возвращает размер изображения
Tag	dword	Возвращает и задаёт дополнительные данные
BufferPtr	pointer to byte	Возвращает указатель на буфер с данными изображения

7.1.1.3. Методы

Create (int, int, EnPixelFormat)	Изменяет размер и формат экземпляра TBitmap
Destroy ()	Освобождает память
SetPixel(dint, dint, TColor)	Устанавливает цвет пикселя по заданным координатам
FromFile(string(255))	Загружает изображение из файла
Save(string(255))	Сохраняет изображение в файл (только в формате BMP)

7.1.1.4. Комментарии

Функциональный блок является вспомогательным для работы с изображениями формата BMP.

7.1.1.5. Пример

Демонстрационный пример: Примеры\AgavaTypes\ПЛК30\Bubbles.project.

7.1.2. Функциональный блок TImage

Изображение.

Пространство имён: AgavaTypes.

7.1.2.1. Определение

```
{attribute 'enable_dynamic_creation'}
function_block TImage
```

7.1.2.2. Свойства

Название	Тип	Описание
PixelFormat	EnPixelFormat	Возвращает формат изображения
Width	int	Возвращает ширину изображения в пикселях
Height	int	Возвращает высоту изображения в пикселях
Size	TSize	Возвращает размер изображения
Tag	dword	Возвращает и задаёт дополнительные данные
BufferPtr	pointer to byte	Возвращает указатель на буфер с данными изображения

7.1.2.3. Методы

FromFile(string(255))	Загружает изображение из файла
Save(string(255))	Сохраняет изображение в файл (только в формате BMP)

7.1.2.4. Комментарии

Функциональный блок является вспомогательным для работы с изображениями. Расширяет возможности визуализации при работе с графическим интерфейсом пользователя без использования компонентов Codesys.

7.1.2.5. Пример

Демонстрационный пример: Примеры\AgavaTypes\ПЛК30\Bubbles.project.

7.1.3. Функциональный блок TByteArray

Динамический массив байт.

Пространство имён: AgavaTypes.

7.1.3.1. Определение

```
function_block TByteArray
```

7.1.3.2. Свойства

Название	Тип	Описание
Bytes	pointer to byte	Возвращает указатель на данные массива
Empty	bool	Возвращает признак того, что массив пустой
Size	uint	Возвращает размер массива

7.1.3.3. Методы

AddByte (byte)	Добавляет байт в массив
AddRange(pointer to byte, uint)	Добавляет указанное количество байт из буфера
AddWord(uint)	Добавляет слово (word) в массив. Порядок байт big-endian, т. е. сначала добавляется младший байт
Clear()	Удаляет элементы массива
GetItem(uint): byte	Возвращает элемент массива с указанным индексом
InsertByte(uint, byte)	Вставляет байт в позицию с указанным индексом
InsertWord(uint, uint)	Вставляет слово (word) в позицию с указанным индексом
RemoveByte(uint)	Удаляет элемент с указанным индексом
RemoveRange(uint, uint)	Удаляет диапазон элементов, начиная с указанной позиции
RemoveWord(uint)	Удаляет элемент типа word (два байта) из указанной позиции

7.1.3.4. Комментарии

Функциональный блок использует динамическую память для автоматического изменения своего размера. Память выделяется частями по 512 байт.

7.1.3.5. Пример

Демонстрационный пример: Примеры\AgavaTypes\ПЛК40\ByteArrayTest.project.

7.1.4. Функциональный блок TPointerArray

Динамический массив указателей.

Пространство имён: AgavaTypes.

7.1.4.1. Определение

```
function_block TPointerArray
```

7.1.4.2. Свойства

Название	Тип	Описание
Pointers	pointer to dword	Возвращает указатель на данные массива
Empty	bool	Возвращает признак того, что массив пустой
Size	uint	Возвращает размер массива

7.1.4.3. Методы

AddItem(dword)	Добавляет указатель в массив
AddRange(pointer to byte, uint)	Добавляет указанное количество указателей из буфера
Clear()	Удаляет элементы массива
GetItem(uint): dword	Возвращает элемент массива с указанным индексом
InsertItem(uint, dword)	Вставляет указатель в позицию с указанным индексом
RemoveAt(uint)	Удаляет элемент с указанным индексом
RemoveRange(uint, uint)	Удаляет диапазон элементов, начиная с указанной позиции

7.1.4.4. Комментарии

Функциональный блок использует динамическую память для автоматического изменения своего размера. Память выделяется частями по 32 байт.

7.1.4.5. Пример

Демонстрационный пример: Примеры\AgavaTypes\ПЛК40\PointerArrayTest.project.

7.1.5. Функциональный блок TList

Односвязный список.

Пространство имён: AgavaTypes.

7.1.5.1. Определение

```
function_block TList
```

7.1.5.2. Свойства

Название	Тип	Описание
Count	int	Возвращает количество элементов в списке

7.1.5.3. Методы

AddItem(pointer to byte): pointer to byte	Добавляет элемент в список
Clear()	Очищает список элементов (память данных элементов не освобождается)
GetItem(int): pointer to byte	Возвращает элемент массива с указанным индексом
InsertItem(int, pointer to byte): pointer to byte	Вставляет элемент в указанную позицию
RemoveAt(int)	Удаляет элемент из указанной позиции (память данных элемента не освобождается)

7.1.5.4. Комментарии

Функциональный блок реализует односвязный список с элементами типа TLinkedItem. ФБ работает с динамической памятью, позволяет добавлять и удалять элементы списка без перераспределения всего объёма памяти, занятой элементами списка.

7.1.5.5. Пример

Демонстрационный пример: Примеры\AgavaTypes\ПЛК40\ListTest.project.

7.1.6. Функциональный блок TJson

Реализует чтение и разбор файла в формате JSON.

Пространство имён: AgavaTypes.

7.1.6.1. Определение

```
function_block TJson extends TList
```

7.1.6.2. Методы

Load(string(255)): dint	Загружает файл в формате JSON
ToBool(string(255), bool): int	Возвращает в переменной value значение типа bool для элемента key. При успешном выполнении метод возвращает 0, иначе -1.
ToInt(string(255), dint): int	Возвращает в переменной value значение типа dint для элемента key. При успешном выполнении метод возвращает 0, иначе -1.
ToFloat (string(255), real): int	Возвращает в переменной value значение типа real для элемента key. При успешном выполнении метод возвращает 0, иначе -1.
ToString (string(255), string(255)): int	Возвращает в переменной value значение типа string для элемента key. При успешном выполнении метод возвращает 0, иначе -1.

7.1.6.3. Комментарии

Функциональный блок позволяет считывать файлы в формате JSON и получать доступ к его элементам различного типа. После загрузки файл сохраняется в динамической памяти во внутреннем списке и обращение к элементам идёт через этот список.

7.1.6.4. Пример

Демонстрационный пример: Примеры\AgavaTypes\ПЛК40\JsonTest.project.

7.1.7. Функциональный блок TLocker

Элемент синхронизации между задачами на основе функционального блока bolt.

Пространство имён: AgavaTypes.

7.1.7.1. Определение

```
function_block TLocker
var_input
    sleepus: ulint := 50;
end_var
```

7.1.7.2. Методы

Lock()	Устанавливает блокировку объекта. Если объект заблокирован, то приостанавливает выполнение задачи до её снятия
UnLock()	Снимает блокировку объекта

7.1.7.3. Комментарии

Функциональный блок реализует простой элемент синхронизации между задачами. Входной параметр sleepus (мкс) задаёт интервал проверки состояния блокировки во время ожидания её снятия.

7.1.7.4. Пример

```
_locker: TLocker;

method public Beep
var_input
      wDuration: word;
end_var

// Блокировка метода Beep() на время работы бипера.
_locker.Lock();

Beeper( xEnable := true, xBeep := true, wDuration := wDuration );

_locker.UnLock();
```

7.1.8. Функциональный блок TLogger

Функциональный блок вывода отладочной информации в потоки (порт, файл, сокет).

Пространство имён: AgavaTypes.

7.1.8.1. Определение

```
function_block TLogger
```

7.1.8.2. Методы

Attach(EnLoggerAdapter, handle)	Задаёт адаптер для вывода в виде его дескриптора
logdebug(string(255), array[0 .. 255] of dword)	Выводит отладочное (DEBUG) сообщение
logerror(string(255), array[0 .. 255] of dword)	Выводит сообщение об ошибке (ERROR)
loginfo(string(255), array[0 .. 255] of dword)	Выводит информационное (INFO) сообщение
logwarn(string(255), array[0 .. 255] of dword)	Выводит предупреждение (WARN)

7.1.8.3. Комментарии

Функциональный блок позволяет выводить текстовую отладочную информацию в порт, файл или сокет. В одном цикле можно вывести строку размером не более 4096 байт.

7.1.8.4. Пример

Демонстрационные примеры:

- Примеры\AgavaTypes\ПЛК40\ByteArrayTest.project;

- Примеры\AgavaTypes\ПЛК40\PointerArrayTest.project.

7.1.9. Функциональный блок TSerial

Реализует обёртку для работы с последовательным соединением.

Пространство имён: AgavaTypes.

7.1.9.1. Определение

```
function_block TSerial
```

7.1.9.2. Свойства

Название	Тип	Описание
Baudrate	COM_Baudrate	Скорость
BufferSize	udint	Размер буфера приёмника
Handle	RTS_IEC_HANDLE	Дескриптор устройства
Parity	COM_Parity	Чётность
Port	COM_Ports	Порт
StopBits	COM_StopBits	Количество стоп-битов

7.1.9.3. Методы

Close(): RTS_IEC_RESULT	Закрывает соединение
Open(): RTS_IEC_RESULT	Открывает соединение
Receive(pointer to byte, udint): RTS_IEC_RESULT	Метод позволяет получать данные
Send(pointer to byte, udint): RTS_IEC_RESULT	Метод позволяет отправить данные

7.1.9.4. Комментарии

Функциональный блок является вспомогательным для работы с последовательным соединением. Перед открытием соединения необходимо настроить его параметры.

7.1.9.5. Пример

Демонстрационный пример: Примеры\AgavaTypes\ПЛК40\ByteArrayTest.project.

7.1.10. Функциональный блок TSocket

Реализует обёртку для работы с сетевым соединением (сокетом).

Пространство имён: AgavaTypes.

7.1.10.1. Определение

```
function_block TSocket
```

7.1.10.2. Свойства

Название	Тип	Описание
Handle	RTS_IEC_HANDLE	Дескриптор устройства

7.1.10.3. Методы

Close(): RTS_IEC_RESULT	Закрывает сокет
Connect(string(16), uint): RTS_IEC_RESULT	Выполняет соединение с сервером по указанному адресу и порту
Create(): RTS_IEC_RESULT	Создаёт сокет
Receive(pointer to byte, dint, dint): RTS_IEC_RESULT	Метод позволяет получать данные
Send(pointer to byte, uint, dint): RTS_IEC_RESULT	Метод позволяет отправить данные

7.1.10.4. Комментарии

Функциональный блок является вспомогательным для работы с сетевым соединением.

7.1.10.5. Пример

Демонстрационный пример: Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExTCPMasterSync.project.

7.1.11. Функциональный блок TVersion

Позволяет вести альтернативную Codesys нумерацию версий.

Пространство имён: AgavaTypes.

7.1.11.1. Определение

```
function_block TVersion
```

7.1.11.2. Свойства

Название	Тип	Описание
AsString	string	Версия в виде строки 'Major.Minor.Revision.Build'
Build	word	Возвращает значение Build
BuildDate	string	Дата сборки
BuildTime	string	Время сборки
Major	word	Возвращает или задаёт значение Major
Minor	word	Возвращает или задаёт значение Minor
Revision	word	Возвращает значение Revision

7.1.11.3. Комментарии

Формат: Major.Minor.Revision.Build (Microsoft), где Revision и Build формируются автоматически в зависимости от текущей даты, начиная с эпохи Windows (01.01.2000).

7.1.11.4. Пример

Демонстрационный пример: Примеры\AgavaUI\ПЛК30\ConsoleMenu.project.

7.1.12. Перечисления

Пространство имён: AgavaTypes.

7.1.12.1. EnLoggerAdapter

Тип устройства журнала.

7.1.12.1.1. Определение

```
{attribute 'qualified_only'}
type EnLoggerAdapter :
(
    laNone := 0,    // не определён
    laSerial,       // последовательная линия
    laFile,         // файл
    laSocket        // сокет
);
end_type
```

7.1.12.2. EnPixelFormat

Формат представления пиксела.

7.1.12.2.1. Определение

```
{attribute 'qualified_only'}
type EnPixelFormat :
(
    Undefined := 0,     // не определён
    Format1bpp,         // 1 бит
    Format24bppRgb,     // 24 бита (3 байта)
    Format32bppRgb      // 32 бита (4 байта)
);
end_type
```

7.1.13. Глобальные константы

Пространство имён: AgavaTypes.

```
var_global constant

    NULL: dword := 0;                           // идентификатор "пустого" параметра
    nullptr: pointer to byte := 0;              // идентификатор указателя, равного нулю

    AGAVA_TARGET_ID_HIWORD: word := 16#169D;    // идентификатор семейства Агава ПЛК

end_var
```

Цвета.

```
var_global constant

    // Стандартные цвета.
    clBlack:    dword := 16#FF000000; // Чёрный
    clMaroon:   dword := 16#FF800000; // Тёмно-красный
    clGreen:    dword := 16#FF008000; // Зелёный
    clOlive:    dword := 16#FF808000; // Оливковый
    clNavy:     dword := 16#FF000080; // Тёмно-синий
    clPurple:   dword := 16#FF800080; // Пурпурный
    clTeal:     dword := 16#FF008080; // Стальной
    clGray:     dword := 16#FF808080; // Серый
    clSilver:   dword := 16#FFC0C0C0; // Серебряный
    clRed:      dword := 16#FFFF0000; // Красный
    clLime:     dword := 16#FF00FF00; // Ярко-зелёный
    clYellow:   dword := 16#FFFFFF00; // Жёлтый
    clBlue:     dword := 16#FF0000FF; // Синий
    clFuchsia:  dword := 16#FFFF00FF; // Фиолетовый
    clAqua:     dword := 16#FF00FFFF; // Бирюзовый
    clWhite:    dword := 16#FFFFFFFF; // Белый

    // Красные тона:
    clIndianRed:    dword := 16#FFCD5C5C;
    clLightCoral:   dword := 16#FFF08080;
    clSalmon:       dword := 16#FFFA8072;
    clDarkSalmon:   dword := 16#FFE9967A;
    clLightSalmon:  dword := 16#FFFFA07A;
    clCrimson:      dword := 16#FFDC143C;
    //clRed:          dword := 16#FFFF0000;
    clFireBrick:    dword := 16#FFB22222;
    clDarkRed:      dword := 16#FF8B0000;

    // Розовые тона:
    clPink:         dword := 16#FFFFC0CB;
    clLightPink:    dword := 16#FFFFB6C1;
    clHotPink:      dword := 16#FFFF69B4;
    clDeepPink:     dword := 16#FFFF1493;
    clMediumVioletRed:  dword := 16#FFC71585;
    clPaleVioletRed:    dword := 16#FFDB7093;

    // Оранжевые тона:
```

```
    //clLightSalmon:  dword := 16#FFFFA07A;
    clCoral:        dword := 16#FFFF7F50;
    clTomato:       dword := 16#FFFF6347;
    clOrangeRed:    dword := 16#FFFF4500;
    clDarkOrange:   dword := 16#FFFF8C00;
    clOrange:       dword := 16#FFFFA500;

    // Жёлтые тона:
    clGold:         dword := 16#FFFFD700;
    //clYellow:       dword := 16#FFFFFF00;
    clLightYellow:  dword := 16#FFFFFFE0;
    clLemonChiffon: dword := 16#FFFFFACD;
    clLightGoldenrodYellow: dword := 16#FFFAFAD2;
    clPapayaWhip:   dword := 16#FFFFEFD5;
    clMoccasin:     dword := 16#FFFFE4B5;
    clPeachPuff:    dword := 16#FFFFDAB9;
    clPaleGoldenrod:    dword := 16#FFEEE8AA;
    clKhaki:        dword := 16#FFF0E68C;
    clDarkKhaki:    dword := 16#FFBDB76B;

    // Фиолетовые тона:
    clLavender:     dword := 16#FFE6E6FA;
    clThistle:      dword := 16#FFD8BFD8;
    clPlum:         dword := 16#FFDDA0DD;
    clViolet:       dword := 16#FFEE82EE;
    clOrchid:       dword := 16#FFDA70D6;
    //clFuchsia:      dword := 16#FFFF00FF;
    clMagenta:      dword := 16#FFFF00FF;
    clMediumOrchid: dword := 16#FFBA55D3;
    clMediumPurple: dword := 16#FF9370DB;
    clBlueViolet:   dword := 16#FF8A2BE2;
    clDarkViolet:   dword := 16#FF9400D3;
    clDarkOrchid:   dword := 16#FF9932CC;
    clDarkMagenta:  dword := 16#FF8B008B;
    //clPurple:       dword := 16#FF800080;
    clIndigo:       dword := 16#FF4B0082;
    clSlateBlue:    dword := 16#FF6A5ACD;
    clDarkSlateBlue:    dword := 16#FF483D8B;

    // Коричневые тона:
    clCornsilk:         dword := 16#FFFFF8DC;
    clBlanchedAlmond:   dword := 16#FFFFEBCD;
    clBisque:           dword := 16#FFFFE4C4;
    clNavajoWhite:      dword := 16#FFFFDEAD;
    clWheat:            dword := 16#FFF5DEB3;
    clBurlyWood:        dword := 16#FFDEB887;
    clTan:              dword := 16#FFD2B48C;
    clRosyBrown:        dword := 16#FFBC8F8F;
    clSandyBrown:       dword := 16#FFF4A460;
    clGoldenrod:        dword := 16#FFDAA520;
    clDarkGoldenRod:    dword := 16#FFB8860B;
    clPeru:             dword := 16#FFCD853F;
    clChocolate:        dword := 16#FFD2691E;
    clSaddleBrown:      dword := 16#FF8B4513;
    clSienna:           dword := 16#FFA0522D;
    clBrown:            dword := 16#FFA52A2A;
```

```
    //clMaroon:           dword := 16#FF800000;

    // Основные цвета:
    //clBlack:    dword := 16#FF000000;
    //clGray:     dword := 16#FF808080;
    //clSilver:   dword := 16#FFC0C0C0;
    //clWhite:    dword := 16#FFFFFFFF;
    //clFuchsia:  dword := 16#FFFF00FF;
    //clPurple:   dword := 16#FF800080;
    //clRed:      dword := 16#FFFF0000;
    //clMaroon:   dword := 16#FF800000;
    //clYellow:   dword := 16#FFFFFF00;
    //clOlive:    dword := 16#FF808000;
    //clLime:     dword := 16#FF00FF00;
    //clGreen:    dword := 16#FF008000;
    //clAqua:     dword := 16#FF00FFFF;
    //clTeal:     dword := 16#FF008080;
    //clBlue:     dword := 16#FF0000FF;
    //clNavy:     dword := 16#FF000080;

    // Зелёные тона:
    clGreenYellow:  dword := 16#FFADFF2F;
    clChartreuse:   dword := 16#FF7FFF00;
    clLawnGreen:    dword := 16#FF7CFC00;
    //clLime:         dword := 16#FF00FF00;
    clLimeGreen:    dword := 16#FF32CD32;
    clPaleGreen:    dword := 16#FF98FB98;
    clLightGreen:   dword := 16#FF90EE90;
    clMediumSpringGreen:    dword := 16#FF00FA9A;
    clSpringGreen:  dword := 16#FF00FF7F;
    clMediumSeaGreen:       dword := 16#FF3CB371;
    clSeaGreen:     dword := 16#FF2E8B57;
    clForestGreen:  dword := 16#FF228B22;
    //clGreen:        dword := 16#FF008000;
    clDarkGreen:    dword := 16#FF006400;
    clYellowGreen:  dword := 16#FF9ACD32;
    clOliveDrab:    dword := 16#FF6B8E23;
    //clOlive:        dword := 16#FF808000;
    clDarkOliveGreen:   dword := 16#FF556B2F;
    clMediumAquamarine: dword := 16#FF66CDAA;
    clDarkSeaGreen:     dword := 16#FF8FBC8F;
    clLightSeaGreen:    dword := 16#FF20B2AA;
    clDarkCyan:     dword := 16#FF008B8B;
    //clTeal:         dword := 16#FF008080;

    // Синие тона:
    //clAqua:             dword := 16#FF00FFFF;
    clCyan:             dword := 16#FF00FFFF;
    clLightCyan:        dword := 16#FFE0FFFF;
    clPaleTurquoise:    dword := 16#FFAFEEEE;
    clAquamarine:       dword := 16#FF7FFFD4;
    clTurquoise:        dword := 16#FF40E0D0;
    clMediumTurquoise:  dword := 16#FF48D1CC;
    clDarkTurquoise:    dword := 16#FF00CED1;
    clCadetBlue:        dword := 16#FF5F9EA0;
    clSteelBlue:        dword := 16#FF4682B4;
```

```
    clLightSteelBlue:   dword := 16#FFB0C4DE;
    clPowderBlue:       dword := 16#FFB0E0E6;
    clLightBlue:        dword := 16#FFADD8E6;
    clSkyBlue:          dword := 16#FF87CEEB;
    clLightSkyBlue:     dword := 16#FF87CEFA;
    clDeepSkyBlue:      dword := 16#FF00BFFF;
    clDodgerBlue:       dword := 16#FF1E90FF;
    clCornflowerBlue:   dword := 16#FF6495ED;
    clMediumSlateBlue:  dword := 16#FF7B68EE;
    clRoyalBlue:        dword := 16#FF4169E1;
    //clBlue:             dword := 16#FF0000FF;
    clMediumBlue:       dword := 16#FF0000CD;
    clDarkBlue:         dword := 16#FF00008B;
    //clNavy:             dword := 16#FF000080;
    clMidnightBlue:     dword := 16#FF191970;

    // Белые тона:
    //clWhite:            dword := 16#FFFFFFFF;
    clSnow:             dword := 16#FFFFFAFA;
    clHoneydew:         dword := 16#FFF0FFF0;
    clMintCream:        dword := 16#FFF5FFFA;
    clAzure:            dword := 16#FFF0FFFF;
    clAliceBlue:        dword := 16#FFF0F8FF;
    clGhostWhite:       dword := 16#FFF8F8FF;
    clWhiteSmoke:       dword := 16#FFF5F5F5;
    clSeashell:         dword := 16#FFFFF5EE;
    clBeige:            dword := 16#FFF5F5DC;
    clOldLace:          dword := 16#FFFDF5E6;
    clFloralWhite:      dword := 16#FFFFFAF0;
    clIvory:            dword := 16#FFFFFFF0;
    clAntiqueWhite:     dword := 16#FFFAEBD7;
    clLinen:            dword := 16#FFFAF0E6;
    clLavenderBlush:    dword := 16#FFFFF0F5;
    clMistyRose:        dword := 16#FFFFE4E1;

    // Серые тона:
    clGainsboro:    dword := 16#FFDCDCDC;
    clLightGrey:    dword := 16#FFD3D3D3;
    clLightGray:    dword := 16#FFD3D3D3;
    //clSilver:       dword := 16#FFC0C0C0;
    clDarkGray:     dword := 16#FFA9A9A9;
    clDarkGrey:     dword := 16#FFA9A9A9;
    //clGray:         dword := 16#FF808080;
    clGrey:         dword := 16#FF808080;
    clDimGray:      dword := 16#FF696969;
    clDimGrey:      dword := 16#FF696969;
    clLightSlateGray:   dword := 16#FF778899;
    clLightSlateGrey:   dword := 16#FF778899;
    clSlateGray:    dword := 16#FF708090;
    clSlateGrey:    dword := 16#FF708090;
    clDarkSlateGray:    dword := 16#FF2F4F4F;
    clDarkSlateGrey:    dword := 16#FF2F4F4F;
    //clBlack:        dword := 16#FF000000;

end_var
```

Таблица цветов.

HTML Имя Цвета	HEX	RGB
Красные тона:		
IndianRed	#CD5C5C	205, 92, 92
LightCoral	#F08080	240, 128, 128
Salmon	#FA8072	250, 128, 114
DarkSalmon	#E9967A	233, 150, 122
LightSalmon	#FFA07A	255, 160, 122
Crimson	#DC143C	220, 20, 60
Red	#FF0000	255, 0, 0
FireBrick	#B22222	178, 34, 34
DarkRed	#8B0000	139, 0, 0
Розовые тона:		
Pink	#FFC0CB	255, 192, 203
LightPink	#FFB6C1	255, 182, 193
HotPink	#FF69B4	255, 105, 180
DeepPink	#FF1493	255, 20, 147
MediumVioletRed	#C71585	199, 21, 133
PaleVioletRed	#DB7093	219, 112, 147
Оранжевые тона:		
LightSalmon	#FFA07A	255, 160, 122
Coral	#FF7F50	255, 127, 80
Tomato	#FF6347	255, 99, 71
OrangeRed	#FF4500	255, 69, 0
DarkOrange	#FF8C00	255, 140, 0
Orange	#FFA500	255, 165, 0
Жёлтые тона:		
Gold	#FFD700	255, 215, 0
Yellow	#FFFF00	255, 255, 0
LightYellow	#FFFFE0	255, 255, 224
LemonChiffon	#FFFACD	255, 250, 205
LightGoldenrodYellow	#FAFAD2	250, 250, 210
PapayaWhip	#FFEFD5	255, 239, 213
Moccasin	#FFE4B5	255, 228, 181
PeachPuff	#FFDAB9	255, 218, 185
PaleGoldenrod	#EEE8AA	238, 232, 170
Khaki	#F0E68C	240, 230, 140
DarkKhaki	#BDB76B	189, 183, 107
Фиолетовые тона:		
Lavender	#E6E6FA	230, 230, 250
Thistle	#D8BFD8	216, 191, 216
Plum	#DDA0DD	221, 160, 221
Violet	#EE82EE	238, 130, 238
Orchid	#DA70D6	218, 112, 214
Fuchsia	#FF00FF	255, 0, 255
Magenta	#FF00FF	255, 0, 255
MediumOrchid	#BA55D3	186, 85, 211
MediumPurple	#9370DB	147, 112, 219
BlueViolet	#8A2BE2	138, 43, 226
DarkViolet	#9400D3	148, 0, 211
DarkOrchid	#9932CC	153, 50, 204
DarkMagenta	#8B008B	139, 0, 139
Purple	#800080	128, 0, 128
Indigo	#4B0082	75, 0, 130
SlateBlue	#6A5ACD	106, 90, 205

DarkSlateBlue	#483D8B	72, 61, 139
Коричневые тона:		
Cornsilk	#FFF8DC	255, 248, 220
BlanchedAlmond	#FFEBCD	255, 235, 205
Bisque	#FFE4C4	255, 228, 196
NavajoWhite	#FFDEAD	255, 222, 173
Wheat	#F5DEB3	245, 222, 179
BurlyWood	#DEB887	222, 184, 135
Tan	#D2B48C	210, 180, 140
RosyBrown	#BC8F8F	188, 143, 143
SandyBrown	#F4A460	244, 164, 96
Goldenrod	#DAA520	218, 165, 32
DarkGoldenRod	#B8860B	184, 134, 11
Peru	#CD853F	205, 133, 63
Chocolate	#D2691E	210, 105, 30
SaddleBrown	#8B4513	139, 69, 19
Sienna	#A0522D	160, 82, 45
Brown	#A52A2A	165, 42, 42
Maroon	#800000	128, 0, 0
Основные цвета:		
Black	#000000	0, 0, 0
Gray	#808080	128, 128, 128
Silver	#C0C0C0	192, 192, 192
White	#FFFFFF	255, 255, 255

Fuchsia	#FF00FF	255, 0, 255
Purple	#800080	128, 0, 128
Red	#FF0000	255, 0, 0
Maroon	#800000	128, 0, 0

Yellow	#FFFF00	205, 92, 92
Olive	#808000	240, 128, 128
Lime	#00FF00	250, 128, 114
Green	#008000	233, 150, 122

Aqua	#00FFFF	205, 92, 92
Teal	#008080	240, 128, 128
Blue	#0000FF	250, 128, 114
Navy	#000080	233, 150, 122

Зелёные тона:		
GreenYellow	#ADFF2F	173, 255, 47
Chartreuse	#7FFF00	127, 255, 0
LawnGreen	#7CFC00	124, 252, 0
Lime	#00FF00	0, 255, 0
LimeGreen	#32CD32	50, 205, 50
PaleGreen	#98FB98	152, 251, 152
LightGreen	#90EE90	144, 238, 144
MediumSpringGreen	#00FA9A	0, 250, 154
SpringGreen	#00FF7F	0, 255, 127
MediumSeaGreen	#3CB371	60, 179, 113
SeaGreen	#2E8B57	46, 139, 87
ForestGreen	#228B22	34, 139, 34
Green	#008000	0, 128, 0
DarkGreen	#006400	0, 100, 0
YellowGreen	#9ACD32	154, 205, 50

OliveDrab	#6B8E23	107, 142, 35
Olive	#808000	128, 128, 0
DarkOliveGreen	#556B2F	85, 107, 47
MediumAquamarine	#66CDAA	102, 205, 170
DarkSeaGreen	#8FBC8F	143, 188, 143
LightSeaGreen	#20B2AA	32, 178, 170
DarkCyan	#008B8B	0, 139, 139
Teal	#008080	0, 128, 128
	Синие тона:	
Aqua	#00FFFF	0, 255, 255
Cyan	#00FFFF	0, 255, 255
LightCyan	#E0FFFF	224, 255, 255
PaleTurquoise	#AFEEEE	175, 238, 238
Aquamarine	#7FFFD4	127, 255, 212
Turquoise	#40E0D0	64, 224, 208
MediumTurquoise	#48D1CC	72, 209, 204
DarkTurquoise	#00CED1	0, 206, 209
CadetBlue	#5F9EA0	95, 158, 160
SteelBlue	#4682B4	70, 130, 180
LightSteelBlue	#B0C4DE	176, 196, 222
PowderBlue	#B0E0E6	176, 224, 230
LightBlue	#ADD8E6	173, 216, 230
SkyBlue	#87CEEB	135, 206, 235
LightSkyBlue	#87CEFA	135, 206, 250
DeepSkyBlue	#00BFFF	0, 191, 255
DodgerBlue	#1E90FF	30, 144, 255
CornflowerBlue	#6495ED	100, 149, 237
MediumSlateBlue	#7B68EE	123, 104, 238
RoyalBlue	#4169E1	65, 105, 225
Blue	#0000FF	0, 0, 255
MediumBlue	#0000CD	0, 0, 205
DarkBlue	#00008B	0, 0, 139
Navy	#000080	0, 0, 128
MidnightBlue	#191970	25, 25, 112
	Белые тона:	
White	#FFFFFF	255, 255, 255
Snow	#FFFAFA	255, 250, 250
Honeydew	#F0FFF0	240, 255, 240
MintCream	#F5FFFA	245, 255, 250
Azure	#F0FFFF	240, 255, 255
AliceBlue	#F0F8FF	240, 248, 255
GhostWhite	#F8F8FF	248, 248, 255
WhiteSmoke	#F5F5F5	245, 245, 245
Seashell	#FFF5EE	255, 245, 238
Beige	#F5F5DC	245, 245, 220
OldLace	#FDF5E6	253, 245, 230
FloralWhite	#FFFAF0	255, 250, 240
Ivory	#FFFFF0	255, 255, 240
AntiqueWhite	#FAEBD7	250, 235, 215
Linen	#FAF0E6	250, 240, 230
LavenderBlush	#FFF0F5	255, 240, 245
MistyRose	#FFE4E1	255, 228, 225
	Серые тона:	
Gainsboro	#DCDCDC	220, 220, 220
LightGrey	#D3D3D3	211, 211, 211
LightGray	#D3D3D3	211, 211, 211

Silver	#C0C0C0	192, 192, 192
DarkGray	#A9A9A9	169, 169, 169
DarkGrey	#A9A9A9	169, 169, 169
Gray	#808080	128, 128, 128
Grey	#808080	128, 128, 128
DimGray	#696969	105, 105, 105
DimGrey	#696969	105, 105, 105
LightSlateGray	#778899	119, 136, 153
LightSlateGrey	#778899	119, 136, 153
SlateGray	#708090	112, 128, 144
SlateGrey	#708090	112, 128, 144
DarkSlateGray	#2F4F4F	47, 79, 79
DarkSlateGrey	#2F4F4F	47, 79, 79
Black	#000000	0, 0, 0

7.2. Библиотека AgavaPlc

Библиотека версии 3.5.10.3 содержит драйвера устройств, входящих в состав ПЛК-30, ПЛК-40 и ПЛК-50.

7.2.1. Функциональный блок TBeeper

Драйвер бипера.

Пространство имён: AgavaPlc.

7.2.1.1. Определение

```
function_block TBeeper
var_input
	xEnable: bool;
	xBeep: bool;
	wDuration: word;
end_var
var_output
	xError: bool;
end_var
```

7.2.1.2. Методы

Beep (uint)	Формирует звуковой сигнал указанной длительности (мс)

7.2.1.3. Комментарии

Функциональный блок предоставляет доступ к биперу ПЛК. Звуковой сигнал формируется по фронту на входе xBeep. Максимальная длительность, заданная параметром wDuration, не должна превышать 2 сек. Это ограничение ОС Linux.

7.2.1.4. Пример

Демонстрационный пример: Примеры\AgavaUI\ПЛК30\ConsoleMenu.project.

7.2.2. Функциональный блок TKeypad

Драйвер кнопок (только для АГАВА ПЛК-30).

Пространство имён: AgavaPlc.

7.2.2.1. Определение

```
function_block TKeypad
var_input
	xEnable: bool;
end_var
var_output
	xError: bool;		// ошибка
	xKeyDown: bool;		// кнопка нажата
	xKeyUp: bool;		// кнопка отпущена
	KeyCode: byte;		// код клавиши
end_var
```

7.2.2.2. Свойства

Название	Тип	Описание
Keys	reference to TKeys	Возвращает ссылку на структуру с состоянием кнопок

7.2.2.3. Комментарии

Функциональный блок предоставляет доступ к клавиатуре ПЛК-30.

7.2.2.4. Пример

Демонстрационный пример: Примеры\AgavaUI\ПЛК30\ConsoleMenu.project.

7.2.3. Функциональный блок TLeds

Драйвер светодиодов (АГАВА ПЛК-30, ПЛК-40, ПЛК-50).

Пространство имён: AgavaPlc.

7.2.3.1. Определение

```
function_block TLeds
```

7.2.3.2. Методы

BackColor(bool, bool, bool)	Устанавливает цвет подсветки индикатора ПЛК-30, используя компоненты RGB.
BackLight(EnBackLightColor)	Устанавливает цвет подсветки индикатора ПЛК-30, используя компоненты RGB.
BackLightLevel(byte)	Устанавливает уровень подсветки экрана ПЛК-40 (1-8)
BackLightOff()	Выключает подсветку индикатора
BackLightOn()	Включает подсветку индикатора белым цветом
Close()	Закрывает устройство
LedOff(EnLeds)	Отключает светодиод
LedOn(EnLeds)	Включает светодиод
LedToggle (EnLeds)	Переключает светодиод
Open()	Открывает устройство

7.2.3.3. Комментарии

Функциональный блок предоставляет доступ к управлению светодиодами ПЛК. Перед использованием управляющих методов необходимо вызвать метод Open() для инициализации блока. Это можно сделать один раз в первом цикле программы. Управление светодиодами осуществляется через файловую систему ОС Linux.

7.2.3.4. Пример

Демонстрационный пример: Примеры\AgavaUI\ПЛК30\ConsoleMenu.project.

7.2.4. Функциональный блок TPlcType

Определение типа контроллера АГАВА.

Пространство имён: AgavaPlc.

7.2.4.1. Определение

```
function_block TPlcType
var_output
    eAgavaType : EnAgavaDevType := EnAgavaDevType.Unknown;      // Тип контроллера
    sDeviceName : string;   // Текстовое представление типа контроллера
end_var
```

7.2.4.2. Комментарии

Функциональный блок определяет тип используемого ПЛК.

7.2.5. Функциональный блок TPowerSensor

Функциональный блок для работы с датчиком сети (только для АГАВА ПЛК-30).

Пространство имён: AgavaPlc.

7.2.5.1. Определение

```
function_block TPowerSensor
var_input
	xEnable: bool;
end_var
var_output
	xError: bool;	// ошибка
	PowerState : EnPowerState := EnPowerState.None; // Код события датчика сети
end_var
```

7.2.5.2. Комментарии

Функциональный блок предоставляет доступ к событиям датчика сети ПЛК-30.

7.2.5.3. Пример

Демонстрационный пример: Примеры\AgavaUI\ПЛК30\ConsoleMenu.project.

7.2.6. Функциональный блок TRetainStorage

Хранение retain и persistent переменных.

Пространство имён: AgavaPlc.

7.2.6.1. Определение

```
function_block TRetainStorage
var_input
      Destination : EnStorageDevice := EnStorageDevice.DevSD;     // Место хранения Retain-
переменных
      SaveFilterTime : time := t#5s;              // Период сохранения
      ForceSave : bool := false;                  // Мгновенное сохранение всех Retain-
переменных (без кэширования).
end_var
var_output
      RTSIECResult : RTS_IEC_RESULT;      // Код состояния
      AttemptSaveCounter: dint;           // Счетчик записей Retain-переменных на носитель
end_var
```

7.2.6.2. Комментарии

Если в приложении используются retain и persistent переменные, то обязательно наличие экземпляра этого функционального блока. Он управляет чтением и записью специального файла, сохраняющего области retain и persistent переменных.

Функциональный блок предоставляет доступ к управлению местом хранения retain и persistent переменных в ПЛК. При первом выполнении функционально блока происходит восстановление значений для областей retain и persistent переменных. Необходимо, чтобы первой командой в приложении осуществлялся вызов этого функционального блока.

Нужно иметь также в виду, что отсчёт интервала времени для сохранения изменений будет вестись только если этот функциональный блок будет постоянно вызываться. Желательно поместить его вызов в отдельную задачу.

Области retain и persistent переменных не записываются, если значения переменных в них не изменились. Это сделано для уменьшения количества циклов записи.

Не рекомендуется хранить в retain и persistent областях счётчики и прочие быстро меняющиеся переменные, т.к. множество циклов записи может значительно уменьшить ресурс используемой для хранения памяти. Также нужно быть внимательным при использовании режима ForceSave.

Объём области памяти retain переменных не имеет ограничения сверху. Минимальный его размер установлен равным 512 байт.

7.2.6.3. Пример

Демонстрационный пример: Проекты\ПЛК-40\Хранитель экрана\ScreenSaver.project.

7.2.7. Функциональный блок TRtc

Драйвер часов реального времени (RTC – real time clock).

Пространство имён: AgavaPlc.

7.2.7.1. Определение

```
function_block TRtc
```

7.2.7.2. Методы

Read: TRtcTime	Считывает время из RTC
Write(TRtcTime)	Устанавливает время в RTC

7.2.7.3. Комментарии

Функциональный блок предоставляет доступ к часам реального времени ПЛК.

7.2.7.4. Пример

Демонстрационный пример: Примеры\AgavaPlc\ПЛК-40\DateTime.project.

7.2.8. Функция Reboot

Перезагрузка ПЛК.

Пространство имён: AgavaPlc.

7.2.8.1. Определение

```
function Reboot: RTS_IEC_RESULT
```

7.2.8.2. Комментарии

Выполняет перезагрузку ПЛК.

7.2.9. Функция Shutdown

Завершение работы ПЛК.

Пространство имён: AgavaPlc.

7.2.9.1. Определение

```
function Shutdown: RTS_IEC_RESULT
```

7.2.9.2. Комментарии

Выполняет завершение работы ПЛК.

7.2.10. Структуры

Пространство имён: AgavaPlc.

7.2.10.1. TInputEvent

Событие.

7.2.10.1.1. Определение

```
type TInputEvent:
struct
	T1: dword;
	T2: dword;
	EvType: word;
	EvCode: word;
	Value: dword;
end_struct
end_type
```

7.2.10.2. TKeys

Состояние кнопок.

7.2.10.2.1. Определение

```
type TKeys:
struct
	Enter:	bool;
	Minus:	bool;
	Comma:	bool;
	Down:	bool;
	Sound:	bool;
	Menu:	bool;
	Start:	bool;
	Stop:	bool;
	Up:	bool;
	Light:	bool;
	Num0:	bool;
	Num1:	bool;
	Num2:	bool;
	Num3:	bool;
	Num4:	bool;
	Num5:	bool;
	Num6:	bool;
	Num7:	bool;
	Num8:	bool;
	Num9:	bool;
end_struct
end_type
```

7.2.10.3. TRtcTime

Дата и время.

7.2.10.3.1. Определение

```
type TRtcTime :
struct
    sec: udint;
    minute: udint;
    hour: udint;
    mday: udint;
    mon: udint;
    year: udint;
    wday: udint;
    yday: udint;
    isdst: udint;
end_struct
end_type
```

7.2.11. Перечисления

Пространство имён: AgavaPlc

7.2.11.1. EnAgavaDevType

Типы контроллеров производства КБ Агава.

7.2.11.1.1. Определение

```
{attribute 'qualified_only'}
{attribute 'strict'}
type EnAgavaDevType :
(
      Unknown := 0,       // Неизвестный тип
      Agava6432_30,       // Агава 6432.30
      Agava6432_40_043,   // Агава 6432.40 4,3" TFT
      Agava6432_40_070,   // Агава 6432.40 7" TFT
      Agava6432_40_100    // Агава 6432.40 10" TFT
);
end_type
```

7.2.11.2. EnBackLightColor

Цвет подсветки (для экрана ПЛК-30).

7.2.11.2.1. Определение

```
{attribute 'qualified_only'}
type EnBackLightColor :
(
    clBlack := 0,
    clWhite,
    clRed,
    clGreen,
    clBlue,
    clYellow,
    clAqua,
    clMagenta
);
end_type
```

7.2.11.3. EnLeds

Светодиоды (Агава ПЛК-30 (40)).

7.2.11.3.1. Определение

```
{attribute 'qualified_only'}
type EnLeds :
(
    ledWork := 0,
    ledWorkUser,
```

```
    ledProgram,
    ledProgUser,
    ledFault,
    ledFaultUser,
    backLightRed,
    backLightGreen,
    backLightBlue
);
end_type
```

7.2.11.4. EnPowerState

Типы состояний датчика сети.

7.2.11.4.1. Определение

```
{attribute 'qualified_only'}
{attribute 'strict'}
type EnPowerState :
(
    None := 0,      // Нет событий по питанию
    PowerSleep,     // Режим сна
    PowerLost,      // Питание потеряно
    PowerResume     // Питание появилось
);
end_type
```

7.2.11.5. EnStorageDevice

Устройство хранения retain-переменных.

7.2.11.5.1. Определение

```
{attribute 'qualified_only'}
{attribute 'strict'}
type EnStorageDevice :
(
      DevSD := 0,               // SD-карта
      DevSysFS                  // Системная файловая система
);
end_type
```

7.2.12. Глобальные константы

Пространство имён: AgavaPlc.

```
var_global constant

// Virtual Keys, AGAVA PLC set.

      VK_ENTER:  byte := 16#0D;
      VK_MINUS:  byte := 16#2D;
      VK_COMMA:  byte := 16#2E;
      VK_DOWN:   byte := 16#64;
      VK_SOUND:  byte := 16#6C;
      VK_MENU:   byte := 16#6D;
      VK_START:  byte := 16#70;
      VK_STOP:   byte := 16#73;
      VK_UP:     byte := 16#75;
      VK_LIGHT:  byte := 16#7A;

// VK_0 - VK_9 are the same as ASCII '0' - '9' (0x30 - 0x39)

      VK_0: byte := 16#30;
      VK_1: byte := 16#31;
      VK_2: byte := 16#32;
      VK_3: byte := 16#33;
      VK_4: byte := 16#34;
      VK_5: byte := 16#35;
      VK_6: byte := 16#36;
      VK_7: byte := 16#37;
      VK_8: byte := 16#38;
      VK_9: byte := 16#39;

end_var
```

7.3. Библиотека AgavaModules

Библиотека версии 3.5.10.1 содержит функциональные блоки, предназначенные для работы с субмодулями ПЛК-40, а также с МВВ-40 по последовательной линии RS-485 или по сети Ethernet. Все функциональные блоки, работающие с МВВ-40, имеют вход дескриптора линии связи. Его значение можно получить как при помощи стандартных библиотек (SysCom, SysSock), так и при помощи вспомогательных функциональных блоков (TSerial, TSocket), входящих в состав SDK. В демонстрационных примерах используется второй способ.

Связь с внутренними субмодулями ПЛК-40 происходит по последовательной локальной шине. Время опроса каждого встраиваемого модуля занимает до 10 мс. Это следует учитывать при выборе времени цикла работы программы, в которой происходит обращение к модулям.

7.3.1. Адресация субмодулей

Обратите внимание! Адресация субмодулей ПЛК-40 и МВВ-40 отличается.

Адресация субмодулей в ПЛК-40 должна использовать глобальные определения SLOT_A … SLOT_F, соответствующие номеру слота.

Адресация субмодулей в МВВ-40 иная. Все субмодули объединяются в группы (AI / AO / DI / DO). Номера субмодулей внутри группы присваиваются в интервале 1…<количество модулей в группе>.

Например, если в МВВ установлены субмодули AI, AI, DI, DO, TMP, DI, то для вызовов соответствующих ФБ в параметр «modno» нужно установить следующие номера субмодулей:

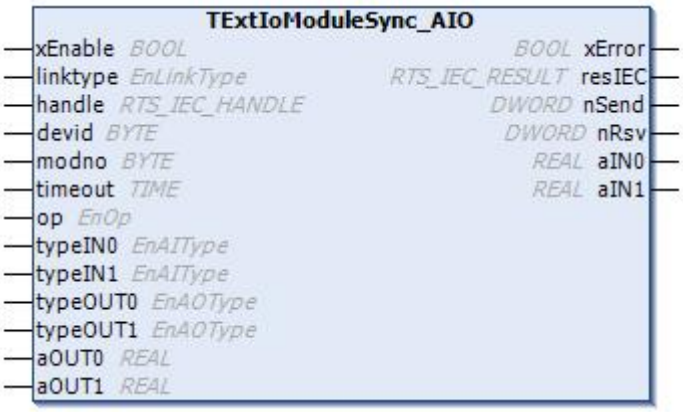
- AI – номера 1, 2
- DI – номера 1, 2
- DO – номер 1
- TMP – номер 1

7.3.2. Функциональный блок TExtIoModuleSync_AIO

Пространство имён: AgavaModules.

7.3.2.1. Определение

```
// Функциональный блок субмодуля AIO, входящего в состав MBB-40.
// 2 аналоговых входа 0-20 мА / 0-10 В
// 2 аналоговых выхода 0-20 мА / 0-10 В
function_block TExtIoModuleSync_AIO extends TExtIoModuleSync
```



7.3.2.2. Входы

Название	Тип	Описание
xEnable	bool	Разрешение работы блока
linktype	EnLinkType	Тип линии
handle	RTS_IEC_HANDLE	Дескриптор линии связи
devid	byte	Адрес устройства (для последовательной линии)
modno	byte	Номер модуля по порядку, начиная с единицы (см. РЭ на MBB-40)
timeout	time	Время ожидания обработки запроса
op	EnOp	Тип операции (чтение/запись)
typeIN0	EnAIType	Тип аналогового входа 1
typeIN1	EnAIType	Тип аналогового входа 2
typeOUT0	EnAOType	Тип аналогового выхода 1
typeOUT1	EnAOType	Тип аналогового выхода 2
aOUT0	real	Аналоговый выход 1 (в мА или В)
aOUT1	real	Аналоговый выход 2 (в мА или В)

7.3.2.3. Выходы

Название	Тип	Описание
xError	bool	Состояние ошибки

aIN0	real	Аналоговый вход 1 (в мА или В)
aIN1	real	Аналоговый вход 2 (в мА или В)
resIEC	RTS_IEC_RESULT	Результат выполнения: ERR_OK - успешно
nSend	dword	Число запросов модуля
nRsv	dword	Число ответов от модуля

7.3.2.4. Комментарии

Функциональный блок предоставляет доступ к данным, настройке и управлению входами и выходами субмодулей МВВ-40, имеющими тип AIO.

7.3.2.5. Примеры

Демонстрационные примеры на языке ST:

Примеры\AgavaModules\ПЛК40\ST\SerialExtIoModuleSync_AIO.project,
Примеры\AgavaModules\ПЛК40\ST\SocketExtIoModuleSync_AIO.project.

Демонстрационные примеры на языке CFC:

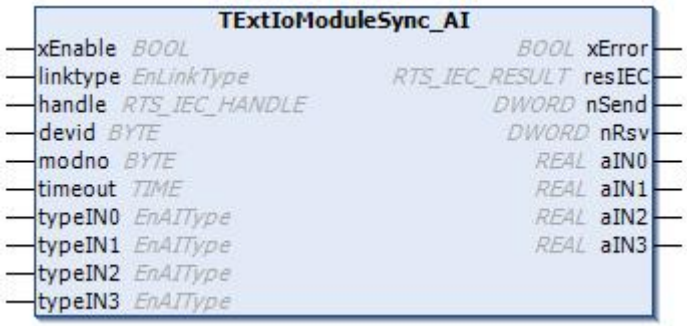
Примеры\AgavaModules\ПЛК40\CFC\SerialExtIoModuleSync_AIO.project,
Примеры\AgavaModules\ПЛК40\CFC\SocketExtIoModuleSync_AIO.project.

7.3.3. Функциональный блок TExtIoModuleSync_AI

Пространство имён: AgavaModules.

7.3.3.1. Определение

```
// Функциональный блок субмодуля AI, входящего в состав МВВ-40.
// 4 аналоговых входа 0-20мА/0-10В
function_block TExtIoModuleSync_AI extends TExtIoModuleSync
```



7.3.3.2. Входы

Название	Тип	Описание
xEnable	bool	Разрешение работы блока
linktype	EnLinkType	Тип линии
handle	RTS_IEC_HANDLE	Дескриптор линии связи
devid	byte	Адрес устройства (для последовательной линии)
modno	byte	Номер модуля по порядку, начиная с единицы (см. РЭ на МВВ-40)
timeout	time	Время ожидания обработки запроса
typeIN0	EnAIType	Тип аналогового входа 1
typeIN1	EnAIType	Тип аналогового входа 2
typeIN2	EnAIType	Тип аналогового входа 3
typeIN3	EnAIType	Тип аналогового входа 4

7.3.3.3. Выходы

Название	Тип	Описание
xError	bool	Состояние ошибки
aIN0	real	Аналоговый вход 1 (в мА или В)
aIN1	real	Аналоговый вход 2 (в мА или В)
aIN2	real	Аналоговый вход 3 (в мА или В)
aIN3	real	Аналоговый вход 4 (в мА или В)
resIEC	RTS_IEC_RESULT	Результат выполнения: ERR_OK - успешно

nSend	dword	Число запросов модуля
nRsv	dword	Число ответов от модуля

7.3.3.4. Комментарии

Функциональный блок предоставляет доступ к данным, настройке и управлению входами субмодулей МВВ-40, имеющими тип AI.

7.3.3.5. Примеры

Демонстрационные примеры на языке ST:

Примеры\AgavaModules\ПЛК40\ST\SerialExtIoModuleSync_AI.project,
Примеры\AgavaModules\ПЛК40\ST\SocketExtIoModuleSync_AI.project.

Демонстрационные примеры на языке CFC:

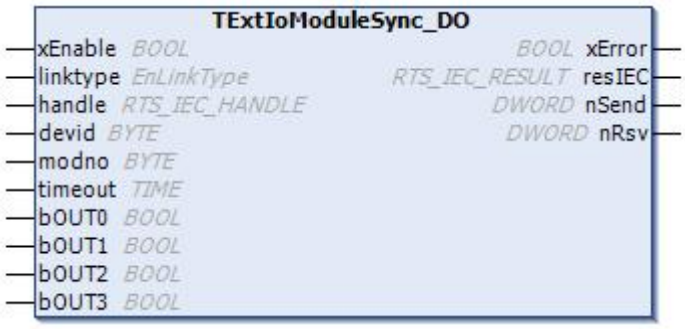
Примеры\AgavaModules\ПЛК40\CFC\SerialExtIoModuleSync_AI.project,
Примеры\AgavaModules\ПЛК40\CFC\SocketExtIoModuleSync_AI.project.

7.3.4. Функциональный блок TExtIoModuleSync_DO

Пространство имён: AgavaModules.

7.3.4.1. Определение

```
// Функциональный блок субмодуля DO, входящего в состав МВВ-40.
// 4 дискретных выхода типа "открытый коллектор"
function_block TExtIoModuleSync_DO extends TExtIoModuleSync
```



7.3.4.2. Входы

Название	Тип	Описание
xEnable	bool	Разрешение работы блока
linktype	EnLinkType	Тип линии
handle	RTS_IEC_HANDLE	Дескриптор линии связи
devid	byte	Адрес устройства (для последовательной линии)
modno	byte	Номер модуля по порядку, начиная с единицы (см. РЭ на МВВ-40)
timeout	time	Время ожидания обработки запроса
bOUT0	bool	Выход ОК 1
bOUT1	bool	Выход ОК 2
bOUT2	bool	Выход ОК 3
bOUT3	bool	Выход ОК 4

7.3.4.3. Выходы

Название	Тип	Описание
xError	bool	Состояние ошибки
resIEC	RTS_IEC_RESULT	Результат выполнения: ERR_OK - успешно
nSend	dword	Число запросов модуля
nRsv	dword	Число ответов от модуля

7.3.4.4. Комментарии

Функциональный блок предоставляет доступ к данным, настройке и управлению выходами субмодулей МВВ-40, имеющими тип DO.

7.3.4.5. Примеры

Демонстрационные примеры на языке ST:

Примеры\AgavaModules\ПЛК40\ST\SerialExtIoModuleSync_DO.project,
Примеры\AgavaModules\ПЛК40\ST\SocketExtIoModuleSync_DO.project.

Демонстрационные примеры на языке CFC:

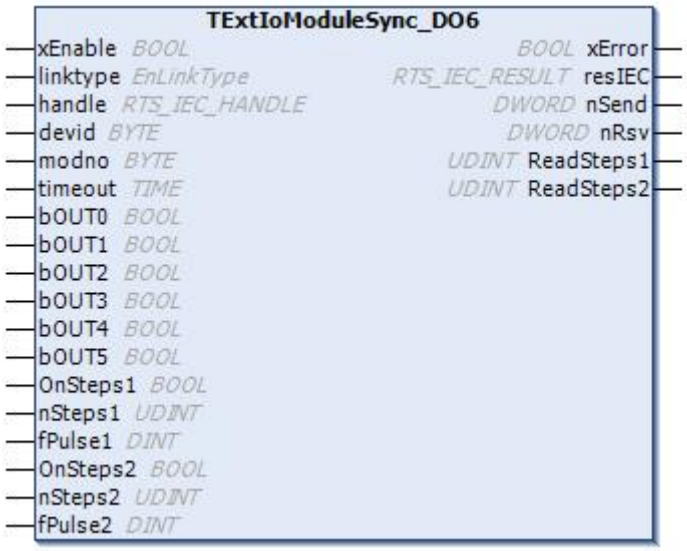
Примеры\AgavaModules\ПЛК40\CFC\SerialExtIoModuleSync_DO.project,
Примеры\AgavaModules\ПЛК40\CFC\SocketExtIoModuleSync_DO.project.

7.3.5. Функциональный блок TExtIoModuleSync_DO6

Пространство имён: AgavaModules.

7.3.5.1. Определение

```
// Функциональный блок субмодуля DO6, входящего в состав МВВ-40.
// 6 дискретных выходов типа "открытый коллектор" или 4 дискретных выхода и управление двумя шаговыми двигателями
function_block TExtIoModuleSync_DO6 extends TExtIoModuleSync
```



7.3.5.2. Входы

Название	Тип	Описание
xEnable	bool	Разрешение работы блока
linktype	EnLinkType	Тип линии
handle	RTS_IEC_HANDLE	Дескриптор линии связи
devid	byte	Адрес устройства (для последовательной линии)
modno	byte	Номер модуля по порядку, начиная с единицы (см. РЭ на МВВ-40)
timeout	time	Время ожидания обработки запроса
bOUT0	bool	Выход ОК 1
bOUT1	bool	Выход ОК 2
bOUT2	bool	Выход ОК 3
bOUT3	bool	Выход ОК 4
bOUT4	bool	Выход ОК 5
bOUT5	bool	Выход ОК 6
OnSteps1	bool	Запуск 1 канала шагового двигателя

nSteps1	udint	Количество шагов 1 канала шагового двигателя
fPulse1	udint	Частота импульсов 1 канала шагового двигателя, в Гц
OnSteps2	bool	Запуск 2 канала шагового двигателя
nSteps2	udint	Количество шагов 2 канала шагового двигателя
fPulse2	udint	Частота импульсов 2 канала шагового двигателя, в Гц

7.3.5.3. Выходы

Название	Тип	Описание
xError	bool	Состояние ошибки
resIEC	RTS_IEC_RESULT	Результат выполнения: ERR_OK - успешно
nSend	dword	Число запросов модуля
nRsv	dword	Число ответов от модуля
ReadSteps1	udint	Количество шагов 1 канала шагового двигателя
ReadSteps2	udint	Количество шагов 2 канала шагового двигателя

7.3.5.4. Комментарии

Функциональный блок предоставляет доступ к данным, настройке и управлению выходами субмодулей МВВ-40, имеющими тип DO-6.

7.3.5.5. Примеры

Демонстрационные примеры на языке ST:

Примеры\AgavaModules\ПЛК40\ST\SerialExtIoModuleSync_DO6.project,
Примеры\AgavaModules\ПЛК40\ST\SocketExtIoModuleSync_DO6.project.

Демонстрационные примеры на языке CFC:

Примеры\AgavaModules\ПЛК40\CFC\SerialExtIoModuleSync_DO6.project,
Примеры\AgavaModules\ПЛК40\CFC\SocketExtIoModuleSync_DO6.project.

Демонстрационные примеры на языке LD:

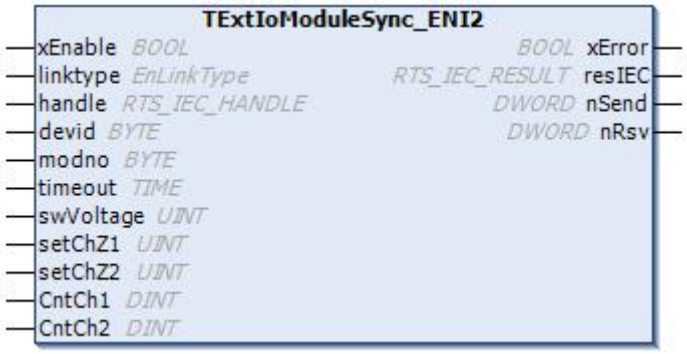
Примеры\AgavaModules\ПЛК40\LD\SerialExtIoModuleSync_DO6.project,
Примеры\AgavaModules\ПЛК40\LD\SocketExtIoModuleSync_DO6.project.

7.3.6. Функциональный блок TExtIoModuleSync_ENI2

Пространство имён: AgavaModules.

7.3.6.1. Определение

```
// Функциональный блок субмодуля энкодера ENI-2, входящего в состав MBB-40.
// 2 счетных канала
function_block TExtIoModuleSync_ENI2 extends TExtIoModuleSync
```



7.3.6.2. Входы

Название	Тип	Описание
xEnable	bool	Разрешение работы блока
linktype	EnLinkType	Тип линии
handle	RTS_IEC_HANDLE	Дескриптор линии связи
devid	byte	Адрес устройства (для последовательной линии)
modno	byte	Номер модуля по порядку, начиная с единицы (см. РЭ на MBB-40)
timeout	time	Время ожидания обработки запроса
swVoltage	uint	Напряжение коммутации (общее для двух каналов): 0 – 5 В 1 – 12 В 2 – 24 В
setChZ1	uint	Разрешение запуска счета при поступлении импульса с z – контакта первого канала: 0 – счет идет постоянно 1 – обнуление счетного канала 2 – была сработка запуска счетчика по z импульсу
setChZ2	uint	Разрешение запуска счета при поступлении импульса с z – контакта второго канала: 0 – счет идет постоянно 1 – обнуление счетного канала 2 – была сработка запуска счетчика по z импульсу

CntCh1	dint	Количестово шагов канала 1
CntCh2	dint	Количестово шагов канала 2

7.3.6.3. Выходы

Название	Тип	Описание
xError	bool	Состояние ошибки
resIEC	RTS_IEC_RESULT	Результат выполнения: ERR_OK - успешно
nSend	dword	Число запросов модуля
nRsv	dword	Число ответов от модуля

7.3.6.4. Комментарии

Функциональный блок предоставляет доступ к данным, настройке и управлению выходами субмодулей МВВ-40, имеющими тип ENI-2.

7.3.6.5. Примеры

Демонстрационные примеры на языке ST:

Примеры\AgavaModules\ПЛК40\ST\SerialExtIoModuleSync_ENI2.project,
Примеры\AgavaModules\ПЛК40\ST\SocketExtIoModuleSync_ENI2.project.

Демонстрационные примеры на языке CFC:

Примеры\AgavaModules\ПЛК40\CFC\SerialExtIoModuleSync_ENI2.project,
Примеры\AgavaModules\ПЛК40\CFC\SocketExtIoModuleSync_ENI2.project.

Демонстрационные примеры на языке LD:

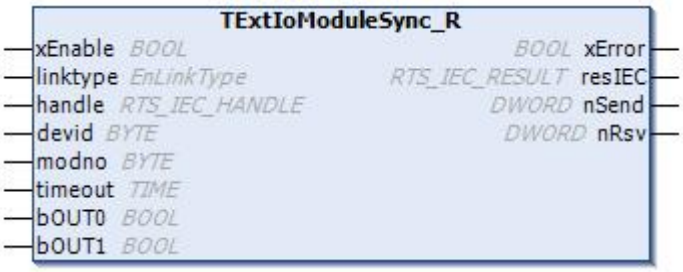
Примеры\AgavaModules\ПЛК40\LD\SerialExtIoModuleSync_ENI2.project,
Примеры\AgavaModules\ПЛК40\LD\SocketExtIoModuleSync_ENI2.project.

7.3.7. Функциональный блок TExtIoModuleSync_R

Пространство имён: AgavaModules.

7.3.7.1. Определение

```
// Функциональный блок субмодуля R, входящего в состав МВВ-40.
// 2 релейных выхода
function_block TExtIoModuleSync_R extends TExtIoModuleSync
```



7.3.7.2. Входы

Название	Тип	Описание
xEnable	bool	Разрешение работы блока
linktype	EnLinkType	Тип линии
handle	RTS_IEC_HANDLE	Дескриптор линии связи
devid	byte	Адрес устройства (для последовательной линии)
modno	byte	Номер модуля по порядку, начиная с единицы (см. РЭ на МВВ-40)
timeout	time	Время ожидания обработки запроса
bOUT0	bool	Выход Реле 1
bOUT1	bool	Выход Реле 2

7.3.7.3. Выходы

Название	Тип	Описание
xError	bool	Состояние ошибки
resIEC	RTS_IEC_RESULT	Результат выполнения: ERR_OK - успешно
nSend	dword	Число запросов модуля
nRsv	dword	Число ответов от модуля

7.3.7.4. Комментарии

Функциональный блок предоставляет доступ к данным, настройке и управлению выходами субмодулей МВВ-40, имеющими тип R.

7.3.7.5. Примеры

Демонстрационные примеры на языке ST:

Примеры\AgavaModules\ПЛК40\ST\SerialExtIoModuleSync_R.project,
Примеры\AgavaModules\ПЛК40\ST\SocketExtIoModuleSync_R.project.

Демонстрационные примеры на языке CFC:

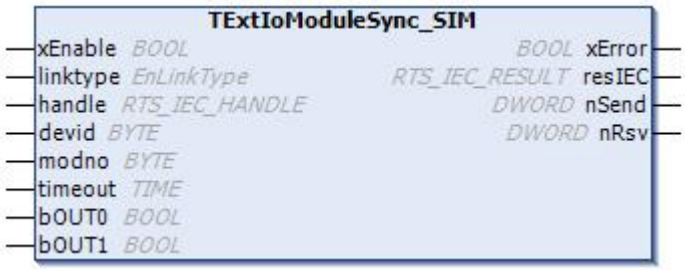
Примеры\AgavaModules\ПЛК40\CFC\SerialExtIoModuleSync_R.project,
Примеры\AgavaModules\ПЛК40\CFC\SocketExtIoModuleSync_R.project.

7.3.8. Функциональный блок TExtIoModuleSync_SIM

Пространство имён: AgavaModules.

7.3.8.1. Определение

```
// Функциональный блок субмодуля SIM, входящего в состав МВВ-40.
// 2 симисторных выхода
function_block TExtIoModuleSync_SIM extends TExtIoModuleSync
```



7.3.8.2. Входы

Название	Тип	Описание
xEnable	bool	Разрешение работы блока
linktype	EnLinkType	Тип линии
handle	RTS_IEC_HANDLE	Дескриптор линии связи
devid	byte	Адрес устройства (для последовательной линии)
modno	byte	Номер модуля по порядку, начиная с единицы (см. РЭ на МВВ-40)
timeout	time	Время ожидания обработки запроса
bOUT0	bool	Выход Симистор 1
bOUT1	bool	Выход Симистор 2

7.3.8.3. Выходы

Название	Тип	Описание
xError	bool	Состояние ошибки
resIEC	RTS_IEC_RESULT	Результат выполнения: ERR_OK - успешно
nSend	dword	Число запросов модуля
nRsv	dword	Число ответов от модуля

7.3.8.4. Комментарии

Функциональный блок предоставляет доступ к данным, настройке и управлению выходами субмодулей МВВ-40, имеющими тип SIM.

7.3.8.5. Примеры

Демонстрационные примеры на языке ST:

Примеры\AgavaModules\External\ПЛК40\ST\SerialExtIoModuleSync_SIM.project,
Примеры\AgavaModules\External\ПЛК40\ST\SocketExtIoModuleSync_SIM.project.

Демонстрационные примеры на языке CFC:

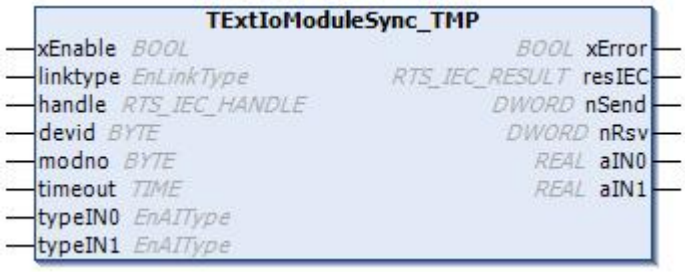
Примеры\AgavaModules\External\ПЛК40\CFC\SerialExtIoModuleSync_SIM.project,
Примеры\AgavaModules\External\ПЛК40\CFC\SocketExtIoModuleSync_SIM.project.

7.3.9. Функциональный блок TExtIoModuleSync_TMP

Пространство имён: AgavaModules.

7.3.9.1. Определение

```
// Функциональный блок субмодуля TMP, входящего в состав MBB-40.
// 2 аналоговых входа термосопротивлений/термопар
function_block TExtIoModuleSync_TMP extends TExtIoModuleSync
```



7.3.9.2. Входы

Название	Тип	Описание
xEnable	bool	Разрешение работы блока
linktype	EnLinkType	Тип линии
handle	RTS_IEC_HANDLE	Дескриптор линии связи
devid	byte	Адрес устройства (для последовательной линии)
modno	byte	Номер модуля по порядку, начиная с единицы (см. РЭ на MBB-40)
timeout	time	Время ожидания обработки запроса
typeIN0	EnAIType	Тип температурного входа 1
typeIN1	EnAIType	Тип температурного входа 2

7.3.9.3. Выходы

Название	Тип	Описание
xError	bool	Состояние ошибки
aIN0	real	Аналоговый вход 1 (Ом, °C, мВ)
aIN1	real	Аналоговый вход 2 (Ом, °C, мВ)
resIEC	RTS_IEC_RESULT	Результат выполнения: ERR_OK - успешно
nSend	dword	Число запросов модуля
nRsv	dword	Число ответов от модуля

7.3.9.4. Комментарии

Функциональный блок предоставляет доступ к данным, настройке и управлению входами субмодулей МВВ-40, имеющими тип TMP.

7.3.9.5. Примеры

Демонстрационные примеры на языке ST:

Примеры\AgavaModules\External\ПЛК40\ST\SerialExtIoModuleSync_TMP.project,
Примеры\AgavaModules\External\ПЛК40\ST\SocketExtIoModuleSync_TMP.project.

Демонстрационные примеры на языке CFC:

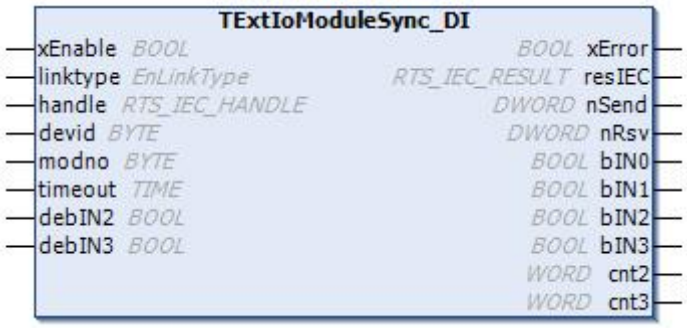
Примеры\AgavaModules\External\ПЛК40\CFC\SerialExtIoModuleSync_TMP.project,
Примеры\AgavaModules\External\ПЛК40\CFC\SocketExtIoModuleSync_TMP.project.

7.3.10. Функциональный блок TExtIoModuleSync_DI

Пространство имён: AgavaModules.

7.3.10.1. Определение

```
// Функциональный блок субмодуля DI, входящего в состав МВВ-40.
// 4 дискретных входа
function_block TExtIoModuleSync_DI extends TExtIoModuleSync
```



7.3.10.2. Входы

Название	Тип	Описание
xEnable	bool	Разрешение работы блока
linktype	EnLinkType	Тип линии
handle	RTS_IEC_HANDLE	Дескриптор линии связи
devid	byte	Адрес устройства (для последовательной линии)
modno	byte	Номер модуля по порядку, начиная с единицы (см. РЭ на МВВ-40)
timeout	time	Время ожидания обработки запроса
debIN2	bool	Антидребезг входа 3
debIN3	bool	Антидребезг входа 4

7.3.10.3. Выходы

Название	Тип	Описание
xError	bool	Состояние ошибки
bIN0	bool	Дискретный вход 1
bIN1	bool	Дискретный вход 2
bIN2	bool	Дискретный вход 3
bIN3	bool	Дискретный вход 4
cnt2	word	Счетчик 3
cnt2	word	Счетчик 4

resIEC	RTS_IEC_RESULT	Результат выполнения: ERR_OK - успешно
nSend	dword	Число запросов модуля
nRsv	dword	Число ответов от модуля

7.3.10.4. Комментарии

Функциональный блок предоставляет доступ к данным, настройке и управлению входами субмодулей МВВ-40, имеющими тип DI.

7.3.10.5. Примеры

Демонстрационные примеры на языке ST:

Примеры\AgavaModules\External\ПЛК40\ST\SerialExtIoModuleSync_DI.project,
Примеры\AgavaModules\External\ПЛК40\ST\SocketExtIoModuleSync_DI.project.

Демонстрационные примеры на языке CFC:

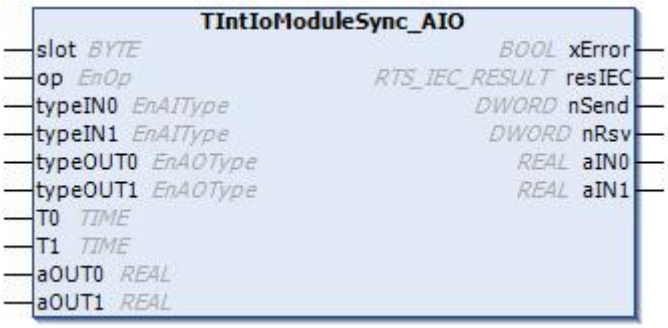
Примеры\AgavaModules\External\ПЛК40\CFC\SerialExtIoModuleSync_DI.project,
Примеры\AgavaModules\External\ПЛК40\CFC\SocketExtIoModuleSync_DI.project.

7.3.11. Функциональный блок TIntIoModuleSync_AIO

Пространство имён: AgavaModules.

7.3.11.1. Определение

```
// Функциональный блок встроенного субмодуля AIO
// 2 аналоговых входа 0-20мА/0-10В
// 2 аналоговых выхода 0-20мА/0-10В
function_block TIntIoModuleSync_AIO extends TIntIoModuleSync
```



7.3.11.2. Входы

Название	Тип	Описание
slot	byte	Номер слота (адрес модуля)
op	EnOp	Тип операции (чтение/запись)
typeIN0	EnAIType	Тип аналогового входа 1
typeIN1	EnAIType	Тип аналогового входа 2
typeOUT0	EnAOType	Тип аналогового выхода 1
typeOUT1	EnAOType	Тип аналогового выхода 2
T0	time	Постоянная времени ФНЧ входа 1
T1	time	Постоянная времени ФНЧ входа 2
aOUT0	real	Аналоговый выход 1 (в мА или В)
aOUT1	real	Аналоговый выход 2 (в мА или В)

7.3.11.3. Выходы

Название	Тип	Описание
xError	bool	Состояние ошибки
resIEC	RTS_IEC_RESULT	Результат выполнения: ERR_OK - успешно
nSend	dword	Число запросов модуля
nRsv	dword	Число ответов от модуля
aIN0	real	Аналоговый вход 1 (в мА или В)
aIN1	real	Аналоговый вход 2 (в мА или В)

7.3.11.4. Комментарии

Функциональный блок предоставляет доступ к субмодулю AIO. Поддерживаемые типы сигналов: 0-10 В и 0-20 мА. Постоянная времени фильтра равная нулю отключает его работу.

7.3.11.5. Пример

Демонстрационный пример на языке ST:

Примеры\AgavaModules\Internal\ПЛК40\ST\TIntIoModuleSync_AIO.project.

Демонстрационный пример на языке CFC:

Примеры\AgavaModules\Internal\ПЛК40\CFC\TIntIoModuleSync_AIO.project.

Демонстрационный пример на языке LD:

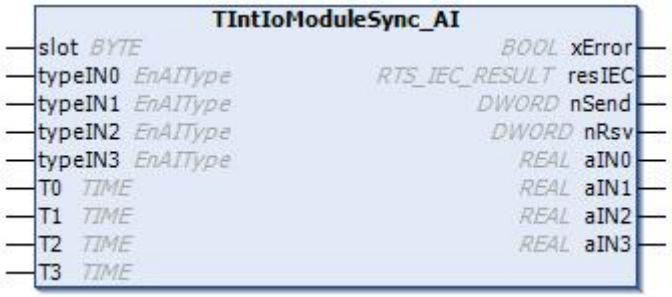
Примеры\AgavaModules\Internal\ПЛК40\LD\TIntIoModuleSync_AIO.project.

7.3.12. Функциональный блок TIntIoModuleSync_AI

Пространство имён: AgavaModules.

7.3.12.1. Определение

```
// Функциональный блок встроенного субмодуля AI
// 4 аналоговых входа 0-20мА/0-10В
function_block TIntIoModuleSync_AI extends TIntIoModuleSync
```



7.3.12.2. Входы

Название	Тип	Описание
slot	byte	Номер слота (адрес модуля)
typeIN0	EnAIType	Тип аналогового входа 1
typeIN1	EnAIType	Тип аналогового входа 2
typeIN2	EnAIType	Тип аналогового входа 3
typeIN3	EnAIType	Тип аналогового входа 4
T0	time	Постоянная времени ФНЧ входа 1
T1	time	Постоянная времени ФНЧ входа 2
T2	time	Постоянная времени ФНЧ входа 3
T3	time	Постоянная времени ФНЧ входа 4

7.3.12.3. Выходы

Название	Тип	Описание
xError	bool	Состояние ошибки
resIEC	RTS_IEC_RESULT	Результат выполнения: ERR_OK - успешно
nSend	dword	Число запросов модуля
nRsv	dword	Число ответов от модуля
aIN0	real	Аналоговый вход 1 (в мА или В)
aIN1	real	Аналоговый вход 2 (в мА или В)
aIN2	real	Аналоговый вход 3 (в мА или В)
aIN3	real	Аналоговый вход 4 (в мА или В)

7.3.12.4. Комментарии

Функциональный блок предоставляет доступ к субмодулю AI. Поддерживаемые типы сигналов: 0-10 В и 0-20 мА. Постоянная времени фильтра равная нулю отключает его работу.

7.3.12.5. Пример

Демонстрационный пример на языке ST:

Примеры\AgavaModules\Internal\ПЛК40\ST\TIntIoModuleSync_AI.project.

Демонстрационный пример на языке CFC:

Примеры\AgavaModules\Internal\ПЛК40\CFC\TIntIoModuleSync_AI.project.

Демонстрационный пример на языке LD:

Примеры\AgavaModules\Internal\ПЛК40\LD\TIntIoModuleSync_AI.project.

7.3.13. Функциональный блок TIntIoModuleSync_DO

Пространство имён: AgavaModules.

7.3.13.1. Определение

```
// Функциональный блок встроенного субмодуля DO
// 4 дискретных выхода типа "открытый коллектор"
function_block TIntIoModuleSync_DO extends TIntIoModuleSync
```

7.3.13.2. Входы

Название	Тип	Описание
slot	byte	Номер слота (адрес модуля)
bOUT0	bool	Выход ОК 1
bOUT1	bool	Выход ОК 2
bOUT2	bool	Выход ОК 3
bOUT3	bool	Выход ОК 4

7.3.13.3. Выходы

Название	Тип	Описание
xError	bool	Состояние ошибки
resIEC	RTS_IEC_RESULT	Результат выполнения: ERR_OK - успешно
nSend	dword	Число запросов модуля
nRsv	dword	Число ответов от модуля

7.3.13.4. Комментарии

Функциональный блок предоставляет доступ к субмодулю DO.

7.3.13.5. Пример

Демонстрационный пример на языке ST:

Примеры\AgavaModules\Internal\ПЛК40\ST\TIntIoModuleSync_DO.project.

Демонстрационный пример на языке CFC:

Примеры\AgavaModules\Internal\ПЛК40\CFC\TIntIoModuleSync_DO.project.

Демонстрационный пример на языке LD:

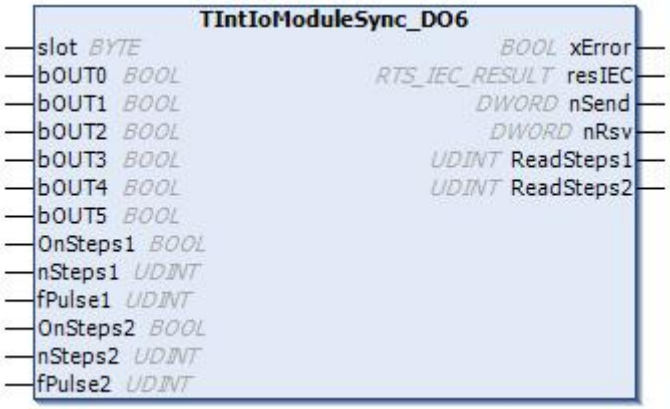
Примеры\AgavaModules\Internal\ПЛК40\LD\TIntIoModuleSync_DO.project.

7.3.14. Функциональный блок TIntIoModuleSync_DO6

Пространство имён: AgavaModules.

7.3.14.1. Определение

```
// Функциональный блок встроенного субмодуля DO-6
// 6 дискретных выходов типа "открытый коллектор" или 4 дискретных выхода и управление двумя шаговыми двигателями.
function_block TIntIoModuleSync_DO6 extends TIntIoModuleSync
```



7.3.14.2. Входы

Название	Тип	Описание
slot	byte	Номер слота (адрес модуля)
bOUT0	bool	Выход ОК 1
bOUT1	bool	Выход ОК 2
bOUT2	bool	Выход ОК 3
bOUT3	bool	Выход ОК 4
bOUT4	bool	Выход ОК 5
bOUT5	bool	Выход ОК 6
OnSteps1	bool	Запуск 1 канала шагового двигателя
nSteps1	udint	Количество шагов 1 канала шагового двигателя
fPulse1	udint	Частота импульсов 1 канала шагового двигателя, в Гц
OnSteps2	bool	Запуск 2 канала шагового двигателя
nSteps2	udint	Количество шагов 2 канала шагового двигателя
fPulse2	udint	Частота импульсов 2 канала шагового двигателя, в Гц

7.3.14.3. Выходы

Название	Тип	Описание
xError	bool	Состояние ошибки
resIEC	RTS_IEC_RESULT	Результат выполнения: ERR_OK - успешно

nSend	dword	Число запросов модуля
nRsv	dword	Число ответов от модуля
ReadSteps1	udint	Количество шагов 1 канала шагового двигателя
ReadSteps2	udint	Количество шагов 2 канала шагового двигателя

7.3.14.4. Комментарии

Функциональный блок предоставляет доступ к субмодулю DO-6.

7.3.14.5. Пример

Демонстрационный пример на языке ST:

Примеры\AgavaModules\Internal\ПЛК40\ST\TIntIoModuleSync_DO6.project.

Демонстрационный пример на языке CFC:

Примеры\AgavaModules\Internal\ПЛК40\CFC\TIntIoModuleSync_DO6.project.

Демонстрационный пример на языке LD:

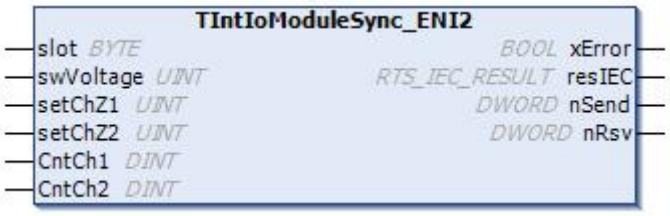
Примеры\AgavaModules\Internal\ПЛК40\LD\TIntIoModuleSync_DO6.project.

7.3.15. Функциональный блок TIntIoModuleSync_ENI2

Пространство имён: AgavaModules.

7.3.15.1. Определение

```
// Функциональный блок встроенного субмодуля энкодера ENI-2
// 2 счетных канала
function_block TIntIoModuleSync_ENI2 extends TIntIoModuleSync
```



7.3.15.2. Входы

Название	Тип	Описание
slot	byte	Номер слота (адрес модуля)
swVoltage	uint	Напряжение коммутации (общее для двух каналов): 0 – 5 В 1 – 12 В 2 – 24 В
setChZ1	uint	Разрешение запуска счета при поступлении импульса с z – контакта первого канала: 0 – счет идет постоянно 1 – обнуление счетного канала 2 – была сработка запуска счетчика по z импульсу
setChZ2	uint	Разрешение запуска счета при поступлении импульса с z – контакта второго канала: 0 – счет идет постоянно 1 – обнуление счетного канала 2 – была сработка запуска счетчика по z импульсу
CntCh1	dint	Количестово шагов канала 1
CntCh2	dint	Количестово шагов канала 2

7.3.15.3. Выходы

Название	Тип	Описание
xError	bool	Состояние ошибки
resIEC	RTS_IEC_RESULT	Результат выполнения: ERR_OK - успешно
nSend	dword	Число запросов модуля
nRsv	dword	Число ответов от модуля

7.3.15.4. Комментарии

Функциональный блок предоставляет доступ к субмодулю ENI-2.

7.3.15.5. Пример

Демонстрационный пример на языке ST:

Примеры\AgavaModules\Internal\ПЛК40\ST\TIntIoModuleSync_ENI2.project.

Демонстрационный пример на языке CFC:

Примеры\AgavaModules\Internal\ПЛК40\CFC\TIntIoModuleSync_ENI2.project.

Демонстрационный пример на языке LD:

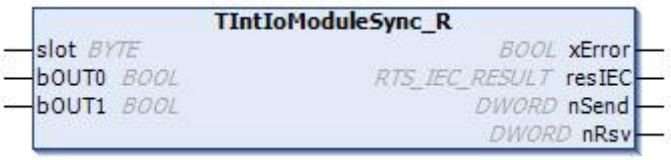
Примеры\AgavaModules\Internal\ПЛК40\LD\TIntIoModuleSync_ENI2.project.

7.3.16. Функциональный блок TIntIoModuleSync_R

Пространство имён: AgavaModules.

7.3.16.1. Определение

```
// Функциональный блок встроенного субмодуля R
// 2 релейных выхода
function_block TIntIoModuleSync_R extends TIntIoModuleSync
```



7.3.16.2. Входы

Название	Тип	Описание
slot	byte	Номер слота (адрес модуля)
bOUT0	bool	Выход Реле 1
bOUT1	bool	Выход Реле 2

7.3.16.3. Выходы

Название	Тип	Описание
xError	bool	Состояние ошибки
resIEC	RTS_IEC_RESULT	Результат выполнения: ERR_OK - успешно
nSend	dword	Число запросов модуля
nRsv	dword	Число ответов от модуля

7.3.16.4. Комментарии

Функциональный блок предоставляет доступ к субмодулю R.

7.3.16.5. Пример

Демонстрационный пример на языке ST:

Примеры\AgavaModules\Internal\ПЛК40\ST\TIntIoModuleSync_R.project.

Демонстрационный пример на языке CFC:

Примеры\AgavaModules\Internal\ПЛК40\CFC\TIntIoModuleSync_R.project.

Демонстрационный пример на языке LD:

Примеры\AgavaModules\Internal\ПЛК40\LD\TIntIoModuleSync_R.project.

7.3.17. Функциональный блок TIntIoModuleSync_SIM

Пространство имён: AgavaModules.

7.3.17.1. Определение

```
// Функциональный блок встроенного субмодуля SIM
// 2 симисторных выхода
function_block TIntIoModuleSync_SIM extends TIntIoModuleSync
```

7.3.17.2. Входы

Название	Тип	Описание
slot	byte	Номер слота (адрес модуля)
bOUT0	bool	Выход Симистор 1
bOUT1	bool	Выход Симистор 2

7.3.17.3. Выходы

Название	Тип	Описание
xError	bool	Состояние ошибки
resIEC	RTS_IEC_RESULT	Результат выполнения: ERR_OK - успешно
nSend	dword	Число запросов модуля
nRsv	dword	Число ответов от модуля

7.3.17.4. Комментарии

Функциональный блок предоставляет доступ к субмодулю SIM.

7.3.17.5. Пример

Демонстрационный пример на языке ST:

Примеры\AgavaModules\Internal\ПЛК40\ST\TIntIoModuleSync_SIM.project.

Демонстрационный пример на языке CFC:

Примеры\AgavaModules\Internal\ПЛК40\CFC\TIntIoModuleSync_SIM.project.

Демонстрационный пример на языке LD:

Примеры\AgavaModules\Internal\ПЛК40\LD\TIntIoModuleSync_SIM.project.

7.3.18. Функциональный блок TIntIoModuleSync_TMP

Пространство имён: AgavaModules.

7.3.18.1. Определение

```
// Функциональный блок встроенного субмодуля TMP
// 2 аналоговых входа термосопротивлений/термопар
function_block TIntIoModuleSync_TMP extends TIntIoModuleSync
```

7.3.18.2. Входы

Название	Тип	Описание
slot	byte	Номер слота (адрес модуля)
typeIN0	EnAIType	Тип температурного входа 1
typeIN1	EnAIType	Тип температурного входа 2
T0	time	Постоянная времени ФНЧ входа 1
T1	time	Постоянная времени ФНЧ входа 2

7.3.18.3. Выходы

Название	Тип	Описание
xError	bool	Состояние ошибки
resIEC	RTS_IEC_RESULT	Результат выполнения: ERR_OK - успешно
nSend	dword	Число запросов модуля
nRsv	dword	Число ответов от модуля
aIN0	real	Аналоговый вход 1 (в Ом, мВ или °С)
aIN1	real	Аналоговый вход 2 (в Ом, мВ или °С)
CJTemp	real	Температура холодного спая (в °С)

7.3.18.4. Комментарии

Функциональный блок предоставляет доступ к субмодулю TMP.

7.3.18.5. Пример

Демонстрационный пример на языке ST:

Примеры\AgavaModules\Internal\ПЛК40\ST\TIntIoModuleSync_TMP.project.

Демонстрационный пример на языке CFC:

Примеры\AgavaModules\Internal\ПЛК40\CFC\TIntIoModuleSync_TMP.project.

Демонстрационный пример на языке LD:

Примеры\AgavaModules\Internal\ПЛК40\LD\TIntIoModuleSync_TMP.project.

7.3.19. Функциональный блок TIntIoModuleSync_DI

Пространство имён: AgavaModules.

7.3.19.1. Определение

```
// Функциональный блок встроенного субмодуля DI
// 4 дискретных входа
function_block TIntIoModuleSync_DI extends TIntIoModuleSync
```

7.3.19.2. Входы

Название	Тип	Описание
slot	byte	Номер слота (адрес модуля)
debIN2	bool	Антидребезг счетчика 3
debIN3	bool	Антидребезг счетчика 4

7.3.19.3. Выходы

Название	Тип	Описание
xError	bool	Состояние ошибки
resIEC	RTS_IEC_RESULT	Результат выполнения: ERR_OK - успешно
nSend	dword	Число запросов модуля
nRsv	dword	Число ответов от модуля
bIN0	bool	Дискретный вход 1
bIN1	bool	Дискретный вход 2
bIN2	bool	Дискретный вход 3
bIN3	bool	Дискретный вход 4
cnt2	word	Счетчик 3 (Дискретный вход 3)
Cnt3	word	Счетчик 4 (Дискретный вход 4)

7.3.19.4. Комментарии

Функциональный блок предоставляет доступ к субмодулю DI.

7.3.19.5. Пример

Демонстрационный пример на языке ST:

Примеры\AgavaModules\Internal\ПЛК40\ST\TIntIoModuleSync_DI.project.

Демонстрационный пример на языке CFC:

Примеры\AgavaModules\Internal\ПЛК40\CFC\TIntIoModuleSync_DI.project.

Демонстрационный пример на языке LD:

Примеры\AgavaModules\Internal\ПЛК40\LD\TIntIoModuleSync_DI.project.

7.3.20. Функциональный блок TIntIoModuleSyncState

Пространство имён: AgavaModules.

7.3.20.1. Определение

```
// Функциональный блок состояния встраиваемого модуля ввода-вывода ПЛК-40.
function_block TIntIoModuleSyncState extends TIntIoModuleSync
```

7.3.20.2. Входы

Название	Тип	Описание
slot	byte	Номер слота (адрес модуля)

7.3.20.3. Выходы

Название	Тип	Описание
xError	bool	Состояние ошибки
resIEC	RTS_IEC_RESULT	Результат выполнения: ERR_OK - успешно
nSend	dword	Число запросов модуля
nRsv	dword	Число ответов от модуля
mType	EnModType	Тип модуля
swVer	word	Версия ПО
nErrRcv	word	Число принятых пакетов с ошибкой
nErrCRC	word	Число ошибок CRC

7.3.20.4. Комментарии

Функциональный блок предоставляет доступ к внутренней информации субмодуля.

7.3.21. Функция ExtIoModulesList

Функция заполняет массив типами установленных в МВВ-40 субмодулей.

Пространство имён: AgavaModules.

7.3.21.1. Определение

```
function ExtIoModulesList: RTS_IEC_RESULT
```

7.3.21.2. Входы

Название	Тип	Описание
linktype	EnLinkType	Тип линии
handle	RTS_IEC_HANDLE	Дескриптор линии связи
devid	byte	Адрес устройства (для последовательной линии)
timeout	time	Время ожидания обработки запроса
list	reference to array [1 .. 6] of int	Массив типов установленных субмодулей

7.3.22. Функция IntIoModulesList

Функция заполняет массив типами установленных в ПЛК-40/ПЛК-60 субмодулей.

Пространство имён: AgavaModules.

7.3.22.1. Определение

```
function IntIoModulesList: RTS_IEC_RESULT
```

7.3.22.2. Входы

Название	Тип	Описание
list	reference to array [1 .. 6] of int	Массив типов установленных субмодулей

7.3.23. Перечисления

Пространство имён: AgavaModules.

7.3.23.1. EnAIType

Тип аналогового входа.

7.3.23.1.1. Определение

```
{attribute 'qualified_only'}
// Тип аналогового входа.
type EnAIType:
(
    st0_10V     :=  0, // 0-10, [В]
    st4_20mA    :=  1, // 4-20, [мА]
    st0_20mA    :=  2, // 0-20, [мА]
    st0_5mA     :=  3, // 0-5, [мА]
    stOhm       :=  4, // Сопротивление, [Ом]
    stPt100     :=  5, // Термосопротивление Pt100, [°C]
    stPt1000    :=  6, // Термосопротивление Pt1000, [°C]
    stTSM50     :=  7, // Термосопротивление ТСМ50, [°C]
    stTSM100    :=  8, // Термосопротивление ТСМ100, [°C]
    stTC_L      :=  9, // Термопара ТХК(L), [°C]
    stTC_J      := 10, // Термопара ТЖК(J), [°C]
    stTC_N      := 11, // Термопара ТНН(N), [°C]
    stTC_K      := 12, // Термопара ТХА(K), [°C]
    stTC_S      := 13, // Термопара ТПП(S), [°C]
    stTC_R      := 14, // Термопара ТПП(R), [°C]
    stTC_B      := 15, // Термопара ТПР(B), [°C]
    stTC_A_1    := 16, // Термопара ТВР(A-1), [°C]
    stTC_A_2    := 17, // Термопара ТВР(A-2), [°C]
    stTC_A_3    := 18, // Термопара ТВР(A-3), [°C]
    stTC_T      := 19, // Термопара ТМК(T), [°C]
    stTSP50     := 20, // Термосопротивление ТСП50, [°C]
    stTSP100    := 21, // Термосопротивление ТСП100, [°C]
    stmV        := 22  // Милливольты, [мВ]
);
end_type
```

7.3.23.2. EnAOType

Тип аналогового выхода.

7.3.23.2.1. Определение

```
{attribute 'qualified_only'}
// Тип аналогового выхода.
type EnAOType:
(
    st0_10V     := 0, // 0-10 В
    st4_20mA    := 1, // 4-20 мА
    st0_20mA    := 2, // 0-20 мА
    st0_5mA     := 3  // 0-5 мА
);
end_type
```

7.3.23.3. EnLinkType

Тип соединения.

7.3.23.3.1. Определение

```
{attribute 'qualified_only'}
// Тип соединения.
type EnLinkType:
(
    ltSerial := 0,  // Последовательная линия (RS-232, RS-485)
    ltSocket := 1   // Линия Ethernet
);
end_type
```

7.3.23.4. EnModType

Типы встраиваемых субмодулей.

7.3.23.4.1. Определение

```
{attribute 'qualified_only'}
// Типы встраиваемых субмодулей МВВ-40.
type EnModType:
(
      mtEmpty := 0,   // Не установлен
      mtDO    := 1,   // Субмодуль DO: 4 дискретных выхода "Общий коллектор"
      mtSIM   := 2,   // Субмодуль SIM: 2 дискретных выхода "Симистор"
      mtR     := 3,   // Субмодуль R: 2 дискретных выхода "Реле"
      mtAI    := 4,   // Субмодуль AI: 4 аналоговых входа 0-20мА/0-10В
      mtAIO   := 5,   // Субмодуль AIO: 2 аналоговых входа 0-20мА/0-10В, 2 аналоговых выхода
0-20мА/0-10В
      mtDI    := 6,   // Субмодуль DI: 4 дискретных входа
      mtTMP   := 7,    // Субмодуль TMP: 2 аналоговых входа для термосопротивлений и
термопар
      mtDO6   := 8,    // Субмодуль DO6: 6 дискретных выходов «Открытый коллектор», либо 4
дискретных выхода «Общий коллектор» и управление двумя шаговыми двигателями
      mtENI2  := 9    // Субмодуль ENI-2: модуль энкодера
);
end_type
```

7.3.23.5. EnOp

Тип операции: чтение/запись.

7.3.23.5.1. Определение

```
{attribute 'qualified_only'}
// Тип операции: чтение/запись.
type EnOp:
(
      Read  := 0, // Чтение
      Write := 1  // Запись
);
```

```
end_type
```

7.3.24. Глобальные константы

Пространство имён: AgavaModules.

7.3.24.1. Slots

```
var_global constant
      SLOT_A: byte := 2;
      SLOT_B: byte := 4;
      SLOT_C: byte := 6;
      SLOT_D: byte := 1;
      SLOT_E: byte := 3;
      SLOT_F: byte := 5;
end_var
```

7.3.24.2. Strings

```
var_global constant

    // Наименования поддерживаемых типов модуля TMP.
    TMP_TYPE_NAMES: array [ 0 .. 18 ] of wstring := [ "Ом", "Pt100", "Pt1000", "TCM50",
"TCM100", "ТХК(L)", "ТЖК(J)", "ТНН(N)", "ТХА(K)", "ТПП(S)", "ТПП(R)", "ТПР(B)", "ТВР(A1)",
"ТВР(A2)", "ТВР(A3)", "ТМК(T)", "ТСП50", "ТСП100", "мВ" ];

end_var
```

7.4. Библиотека AgavaModbus

Библиотека версии 3.5.10.0 содержит реализацию протокола Modbus Master и Slave для режимов RTU и TCP. Работа с регистрами ведётся с использованием обобщённых типов TTag и TModbusRequest.

7.4.1. Функциональный блок TModbusRequest

Пространство имён: AgavaModbus.

7.4.1.1. Определение

```
{attribute 'enable_dynamic_creation'}
function_block TModbusRequest
```

7.4.1.2. Свойства

Название	Тип	Описание
Count	word	Количество регистров в запросе
DeviceId	byte	Номер устройства
FunctionId	byte	Код функции
SendTime	ulint	Время отправки запроса (time_t)
ReceiveTime	ulint	Время приёма запроса (time_t)
RequestMode	EnRequestMode	Режим запроса
RTUBytesExpected	byte	Ожидаемый размер ответа в байтах для Modbus RTU
TCPBytesExpected	byte	Ожидаемый размер ответа в байтах для Modbus TCP
StartAddress	word	Адрес начала запроса
StopAddress	word	Адрес конца запроса
Tags	reference to TList	Список тегов

7.4.1.3. Методы

AddTag(reference to TTag)	Добавляет тег в запрос
Reset()	Сбрасывает состояние запроса

7.4.1.4. Комментарии

Функциональный блок является вспомогательным для работы с регистрами Modbus. С его помощью можно объединять регистры, отличающиеся своим номером при одинаковых значениях номера устройства, кода функции и режима записи. Используется в функциональных блоках TModbusRTUMaster и TModbusTCPMaster при формировании запросов.

Свойства запроса определяются первым добавляемым тегом. Если первый добавляемый тег имеет режим запроса EnRequestMode.rmReadWrite, то перед добавлением тега нужно предварительно установить режим запроса для самого запроса: EnRequestMode.rmRead (чтение) или EnRequestMode.rmWrite (запись).

7.4.1.5. Пример

Демонстрационный пример: Примеры\AgavaModbus\ПЛК40\ModbusRTUMaster.project.

7.4.2. Функциональный блок TModbusRTUMaster

Пространство имён: AgavaModbus.

7.4.2.1. Определение

```
function_block TModbusRTUMaster implements ISerialSettings, IModbusRTUStatistics
```

7.4.2.2. Свойства

Название	Тип	Описание
Port	COM_Ports	Номер последовательного порта
Baudrate	COM_Baudrate	Скорость
Parity	COM_Parity	Чётность
StopBits	COM_StopBits	Количество стоп-битов
CRCErrCount	udint	Количество ошибок CRC
RecvCount	udint	Количество принятых пакетов
SendCount	udint	Количество переданных пакетов
InData	reference to TByteArray	Приёмный буфер
OutData	reference to TByteArray	Передающий буфер
InputRequests	reference to TList	Запросы чтения
OutputRequests	reference to TList	Запросы записи
MaxErrorsCount	uint	Допустимое количество последовательных ошибок
MaxReqRegs	uint	Максимальное количество регистров в запросе

7.4.2.3. Методы

AddTag(pointer to TTag)	Добавляет тег в запрос
ClearRequests()	Удаляет запросы чтения и записи
Open()	Открывает соединение
Close()	Закрывает соединение
Send(reference to TModbusRequest)	Посылает данные запроса
Receive(pointer to byte, udint)	Принимает данные в буфер
MakeRequest(reference to TModbusRequest, udint)	Посылает запрос и ожидает ответа
Process(reference to TModbusRequest)	Выполняет обработку данных из приёмного буфера
ResetStatCounters()	Сбрасывает счётчики статистики обмена

7.4.2.4. Комментарии

Функциональный блок предназначен для реализации протокола Modbus RTU, работающего в режиме клиента (master) на последовательной линии. Работа с регистрами организована через теги (TTag) и

запросы (TModbusRequest). Методы позволяют организовать обмен различными способами в зависимости от решаемой задачи.

7.4.2.5. Пример

Демонстрационный пример: Примеры\AgavaModbus\ПЛК40\ModbusRTUMaster.project.

7.4.3. Функциональный блок TModbusRTUSlave

Пространство имён: AgavaModbus

7.4.3.1. Определение

```
function_block TModbusRTUSlave implements ISerialSettings
var_input
    xEnable: bool;
end_var
var_output
    xError: bool;   // ошибка
    resIEC: RTS_IEC_RESULT;
end_var
```

7.4.3.2. Свойства

Название	Тип	Описание
DeviceId	byte	Адрес устройства
Port	COM_Ports	Номер последовательного порта
Baudrate	COM_Baudrate	Скорость
Parity	COM_Parity	Чётность
StopBits	COM_StopBits	Количество стоп-битов

7.4.3.3. Методы

OnFailure(RTS_IEC_RESULT)	Метод-обработчик ошибки связи
OnReadCoils(pointer to byte, uint, uint, byte)	Метод-обработчик функции READ_COILS
OnReadDiscreteInputs(pointer to byte, uint, uint, byte)	Метод-обработчик функции READ_DISCRETE_INPUTS
OnReadHoldingRegisters(pointer to byte, uint, uint, byte)	Метод-обработчик функции READ_HOLDING_REGISTERS
OnReadInputRegisters(pointer to byte, uint, uint, byte)	Метод-обработчик функции READ_INPUT_REGISTERS
OnWriteMultipleCoils(uint, uint, pointer to byte, byte)	Метод-обработчик функции WRITE_MULTIPLE_COILS
OnWriteMultipleRegisters(uint, uint, pointer to byte, byte)	Метод-обработчик функции WRITE_MULTIPLE_REGISTERS
OnWriteSingleCoil(uint, bool, byte)	Метод-обработчик функции WRITE_SINGLE_COIL
OnWriteSingleRegister(uint, uint, byte)	Метод-обработчик функции WRITE_SINGLE_REGISTER

7.4.3.4. Комментарии

Функциональный блок предназначен для реализации протокола Modbus RTU, работающего в режиме сервера (slave) на последовательной линии. ФБ предоставляет набор методов-обработчиков, упрощающих

реализацию протокола. Перед использованием ФБ нужно настроить параметры последовательного соединения и задать адрес устройства.

7.4.3.5. Пример

Демонстрационный пример: Примеры\AgavaModbus\ПЛК40\ModbusRTUSlave.project.

7.4.4. Функциональный блок TModbusTCPMaster

Пространство имён: AgavaModbus.

7.4.4.1. Определение

```
function_block TModbusTCPMaster implements IModbusTCPStatistics
```

7.4.4.2. Свойства

Название	Тип	Описание
IpAddress	string(16)	IP адрес сервера
Port	uint	Порт сервера
InData	reference to TByteArray	Приёмный буфер
OutData	reference to TByteArray	Передающий буфер
InputRequests	reference to TList	Запросы чтения
OutputRequests	reference to TList	Запросы записи
MaxErrorsCount	uint	Допустимое количество последовательных ошибок
MaxReqRegs	uint	Максимальное количество регистров в запросе
RecvCount	udint	Количество принятых пакетов
SendCount	udint	Количество переданных пакетов

7.4.4.3. Методы

AddTag(pointer to TTag)	Добавляет тег в запрос
ClearRequests()	Удаляет запросы чтения и записи
Connect()	Выполняет соединение с сервером
Close()	Закрывает соединение
MakeRequest(reference to TModbusRequest, udint)	Посылает запрос и ожидает ответа
Process(reference to TModbusRequest)	Выполняет обработку данных из приёмного буфера
Send(reference to TModbusRequest)	Посылает данные запроса
Receive(pointer to byte, udint)	Принимает данные в буфер
ResetStatCounters()	Сбрасывает счётчики статистики обмена

7.4.4.4. Комментарии

Функциональный блок предназначен для реализации протокола Modbus TCP, работающего в режиме клиента (master) по сети Ethernet. Работа с регистрами организована через теги (TTag) и запросы (TModbusRequest). Методы позволяют организовать обмен различными способами в зависимости от решаемой задачи.

7.4.4.5. Пример

Демонстрационный пример: Примеры\AgavaModbus\ПЛК40\ModbusTCPMaster.project.

7.4.5. Функциональный блок TModbusTCPSlave

Пространство имён: AgavaModbus.

7.4.5.1. Определение

```
function_block TModbusTCPSlave
var_input
    xEnable: bool;
end_var
var_output
    xError: bool;   // ошибка
    resIEC: RTS_IEC_RESULT;
end_var
```

7.4.5.2. Свойства

Название	Тип	Описание
DeviceId	byte	Адрес устройства
Port	uint	Порт
MaxConnections	dint	Максимальне количество поддерживаемых соединений

7.4.5.3. Методы

OnFailure(RTS_IEC_RESULT)	Метод-обработчик ошибки связи
OnReadCoils(pointer to byte, uint, uint, byte)	Метод-обработчик функции READ_COILS
OnReadDiscreteInputs(pointer to byte, uint, uint, byte)	Метод-обработчик функции READ_DISCRETE_INPUTS
OnReadHoldingRegisters(pointer to byte, uint, uint, byte)	Метод-обработчик функции READ_HOLDING_REGISTERS
OnReadInputRegisters(pointer to byte, uint, uint, byte)	Метод-обработчик функции READ_INPUT_REGISTERS
OnWriteMultipleCoils(uint, uint, pointer to byte, byte)	Метод-обработчик функции WRITE_MULTIPLE_COILS
OnWriteMultipleRegisters(uint, uint, pointer to byte, byte)	Метод-обработчик функции WRITE_MULTIPLE_REGISTERS
OnWriteSingleCoil(uint, bool, byte)	Метод-обработчик функции WRITE_SINGLE_COIL
OnWriteSingleRegister(uint, uint, byte)	Метод-обработчик функции WRITE_SINGLE_REGISTER

7.4.5.4. Комментарии

Функциональный блок предназначен для реализации протокола Modbus TCP, работающего в режиме сервера (slave) на последовательной линии. ФБ предоставляет набор методов-обработчиков, упрощающих реализацию протокола. Перед использованием ФБ нужно настроить параметры последовательного соединения и задать адрес устройства.

7.4.5.5. Пример

Демонстрационный пример: Примеры\AgavaModbus\ПЛК40\ModbusTCPSlave.project.

7.4.6. Функциональный блок TTag

Пространство имён: AgavaModbus.

7.4.6.1. Определение

```
{attribute 'enable_dynamic_creation'}
function_block TTag
```

7.4.6.2. Свойства

Название	Тип	Описание
Name	string(255)	Задаёт или возвращает имя тега
ShortName	string(255)	Задаёт или возвращает сокращённое имя тега
Description	string(255)	Задаёт или возвращает описание тега
DeviceId	byte	Задаёт или возвращает адрес устройства
ReadFunction	byte	Задаёт или возвращает код функции чтения
ReadRegister	word	Задаёт или возвращает номер регистра для чтения
WriteFunction	byte	Задаёт или возвращает код функции записи
WriteRegister	word	Задаёт или возвращает номер регистра для записи
RequestMode	EnRequestMode	Задаёт или возвращает режим запроса
TagType	EnTagType	Задаёт или возвращает тип тега
AsBool	bool	Задаёт или возвращает значение тега для типа bool
AsByte	byte	Задаёт или возвращает значение тега для типа byte
AsFloat	real	Задаёт или возвращает значение тега для типа real
AsInt	dint	Задаёт или возвращает значение тега для типа dint
AsSByte	sint	Задаёт или возвращает значение тега для типа sint
AsShort	int	Задаёт или возвращает значение тега для типа int
AsUInt	udint	Задаёт или возвращает значение тега для типа udint
AsUShort	uint	Задаёт или возвращает значение тега для типа uint
Length	byte	Возвращает размер тега в регистрах modbus
State	EnTagState	Возвращает состояние тега
ReceiveTime	ulint	Возвращает время приёма данных
Ptr	pointer to byte	Возвращает указатель на буфер, содержащий значение тега

7.4.6.3. Комментарии

Функциональный блок предназначен для представления регистра modbus в виде объекта, содержащего необходимую информацию для автоматизации работы по протоколу. С его помощью описание регистров оформляется в виде текста на языке ST.

7.4.6.4. Пример

Демонстрационные примеры:

- Примеры\IoDrvAgavaLinks\ПЛК40\LinksTest.project;
- Примеры\AgavaModbus\ПЛК40\ModbusRTUMaster.project;
- Примеры\AgavaModbus\ПЛК40\ModbusTCPMaster.project.

7.4.7. Интерфейсы

7.4.7.1. Интерфейс IModbusRTUStatistics

Пространство имён: AgavaModbus.

7.4.7.1.1. Определение

```
interface IModbusRTUStatistics
```

7.4.7.1.2. Свойства

Название	Тип	Описание
CRCErrCount	udint	Количество ошибок CRC
RecvCount	udint	Количество принятых пакетов
SendCount	udint	Количество переданных пакетов

7.4.7.1.3. Методы

ResetStatCounters()	Сбрасывает счётчики статистики обмена

7.4.7.1.4. Комментарии

Предназначен для предоставления значений счётчиков статистики обмена и их сброса.

7.4.7.2. Интерфейс IModbusTCPStatistics

Пространство имён: AgavaModbus.

7.4.7.2.1. Определение

```
interface IModbusTCPStatistics
```

7.4.7.2.2. Свойства

Название	Тип	Описание
RecvCount	udint	Количество принятых пакетов
SendCount	udint	Количество переданных пакетов

7.4.7.2.3. Методы

ResetStatCounters()	Сбрасывает счётчики статистики обмена

7.4.7.2.4. Комментарии

Предназначен для предоставления значений счётчиков статистики обмена и их сброса.

7.4.7.3. Интерфейс ISerialSettings

Пространство имён: AgavaModbus.

7.4.7.3.1. Определение

```
interface ISerialSettings
```

7.4.7.3.2. Свойства

Название	Тип	Описание
Baudrate	COM_Baudrate	Скорость
Parity	COM_Parity	Чётность
Port	COM_Ports	Номер последовательного порта
StopBits	COM_StopBits	Количество стоп-битов

7.4.7.3.3. Комментарии

Предназначен для чтения и записи свойств последовательного соединения.

7.4.8. Перечисления

Пространство имён: AgavaModbus.

7.4.8.1. EnRequestMode

Режим запроса.

7.4.8.1.1. Определение

```
{attribute 'qualified_only'}
{attribute 'strict'}
type EnRequestMode :
(
	rmRead := 0,	// чтение
	rmWrite,	// запись
	rmReadWrite	// чтение и запись
);
end_type
```

7.4.8.2. EnTagState

Состояние тега.

7.4.8.2.1. Определение

```
{attribute 'qualified_only'}
{attribute 'strict'}
type EnTagState :
(
	tsUnknown := 0,	// неизвестно
	tsSuccess,	// ответ получен
	tsTimeout,	// ответ не получен за время ожидания
	tsFailure	// ошибка соединения или обмена
);
end_type
```

7.4.8.3. EnTagType

Тип тега.

7.4.8.3.1. Определение

```
{attribute 'qualified_only'}
{attribute 'strict'}
type EnTagType :
(
	ttUnknown := 0,	// тип не определён
	ttBool,	// тип bool
	ttByte,	// тип byte (не реализован)
	ttSByte,	// тип sint (не реализован)
	ttUShort,	// тип uint
	ttShort,	// тип int
	ttUInt,	// тип udint (не реализован)
	ttInt,	// тип dint (не реализован)
	ttULong,	// тип ulint (не реализован)
	ttLong,	// тип lint (не реализован)
```

```
      ttFloat,            // тип real
      ttDouble            // тип lreal (не реализован)
);
end_type
```

7.4.9. Глобальные константы

Пространство имён: AgavaModbus.

```
var_global constant
      READ_COILS               : byte := 16#01;
      READ_DISCRETE_INPUTS     : byte := 16#02;
      READ_HOLDING_REGISTERS   : byte := 16#03;
      READ_INPUT_REGISTERS     : byte := 16#04;
      WRITE_SINGLE_COIL        : byte := 16#05;
      WRITE_SINGLE_REGISTER    : byte := 16#06;
      WRITE_MULTIPLE_COILS     : byte := 16#0F;
      WRITE_MULTIPLE_REGISTERS : byte := 16#10;
end_var
```

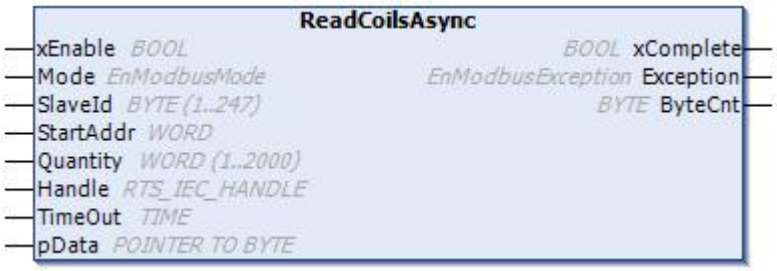
7.5. Библиотека AgavaModbusEx

Библиотека версии 3.5.10.0 содержит набор синхронных и асинхронных функций, реализующих протокол Modbus Master для режимов ASCII, RTU и TCP.

7.5.1. Функциональный блок ReadCoilsAsync

Функция 01_{10} (01_{16}) – Read Coils (асинхронная).

Пространство имён: AgavaModbusEx.



7.5.1.1. Определение

```
function_block ReadCoilsAsync
var_input
    xEnable: bool;              // разрешение работы блока
    Mode: EnModbusMode;         // протокол Modbus
    SlaveId: byte (1..247);     // адрес подчинённого устройства в сети MODBUS
    StartAddr: word;            // начальный адрес дискретных выходов
    Quantity: word (1..2000);   // количество выходов для чтения
    Handle: RTS_IEC_HANDLE;     // дескриптор соединения
    TimeOut: time;              // время тайм-аута [мс] – макс. задержка на обработку
запроса
    pData: pointer to byte;     // байтовый буфер данных
end_var
var_output
    xComplete: bool;            // если = TRUE, то обмен завершен
    Exception: EnModbusException;   // исключения протокола MODBUS или 0xFF – таймаут
    ByteCnt: byte;              // размер прочитанного блока данных, байт
end_var
```

7.5.1.2. Комментарии

Этот ФБ используется для чтения значений дискретных выходов (DO). В запросе PDU задается начальный адрес первого регистра DO и последующее количество необходимых значений DO. В PDU значения DO адресуются, начиная с нуля. Значения DO в ответе находятся в одном байте и соответствуют значению битов. Значения битов определяются как 1 = ON и 0 = OFF. Младший бит первого байта данных содержит значение DO, адрес которого указывался в запросе. Остальные значения DO следуют по нарастающей к старшему значению байта. Если запрашиваемое количество не укладывается в целое число байт, то оставшиеся биты в ответе будут заполнены нулями (в направлении от младшего к старшему байту). Переменная ByteCnt указывает количество полных байтов данных в ответе.

7.5.1.3. Пример

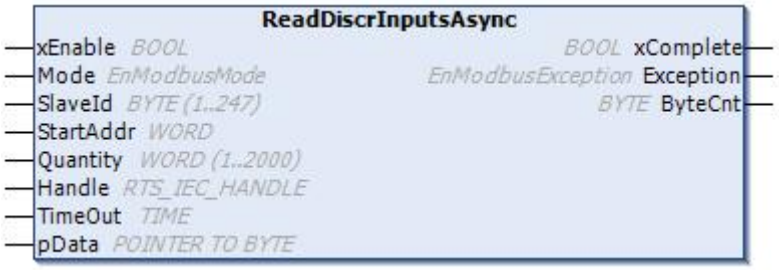
Демонстрационные примеры:

- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExASCIIMasterAsync.project;
- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExRTUMasterAsync.project;
- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExTCPMasterAsync.project.

7.5.2. Функциональный блок ReadDiscrInputsAsync

Функция 02_{10} (02_{16}) – Read Discrete Inputs (асинхронная).

Пространство имён: AgavaModbusEx.



7.5.2.1. Определение

```
function_block ReadDiscrInputsAsync
var_input
    xEnable: bool;              // разрешение работы блока
    Mode: EnModbusMode;         // протокол Modbus
    SlaveId: byte (1..247);     // адрес подчинённого устройства в сети MODBUS
    StartAddr: word;            // начальный адрес дискретных входов
    Quantity: word (1..2000);   // количество входов для чтения
    Handle: RTS_IEC_HANDLE;     // дескриптор соединения
    TimeOut: time;              // время тайм-аута [мс] – макс. задержка на обработку
запроса
    pData: pointer to byte;     // байтовый буфер данных
end_var
var_output
    xComplete: bool;            // если = TRUE, то обмен завершен
    Exception: EnModbusException;   // исключения протокола MODBUS или 0xFF - таймаут
    ByteCnt: byte;              // размер прочитанного блока данных, байт
end_var
```

7.5.2.2. Комментарии

Этот ФБ используется для чтения значений дискретных входов (DI). В запросе PDU задается начальный адрес первого регистра DI и последующее количество необходимых значений DI. В PDU значения DI адресуются, начиная с нуля. Значения DI в ответе находятся в одном байте и соответствуют значению битов. Значения битов определяются как 1 = ON и 0 = OFF. Младший бит первого байта данных содержит значение DI, адрес которого указывался в запросе. Остальные значения DI следуют по нарастающей к старшему значению байта. Если запрашиваемое количество не укладывается в целое число байт, то оставшиеся биты в ответе будут заполнены нулями (в направлении от младшего к старшему байту). Переменная ByteCnt указывает количество полных байтов данных в ответе.

7.5.2.3. Пример

Демонстрационные примеры:

- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExASCIIMasterAsync.project;
- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExRTUMasterAsync.project;

- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExTCPMasterAsync.project.

7.5.3. Функциональный блок ReadHoldRegsAsync

Функция 03_{10} (03_{16}) – Read Holding Registers (асинхронная).

Пространство имён: AgavaModbusEx.

7.5.3.1. Определение

```
function_block ReadHoldRegsAsync
var_input
    xEnable: bool;              // разрешение работы блока
    Mode: EnModbusMode;         // протокол Modbus
    SlaveId: byte (1..247);     // адрес подчинённого устройства в сети MODBUS
    StartAddr: word;            // начальный адрес регистра
    Quantity: byte (1..125);    // количество регистров для чтения
    Handle: RTS_IEC_HANDLE;     // дескриптор соединения
    TimeOut: time;              // время тайм-аута [мс] – макс. задержка на обработку
запроса
    pData: pointer to byte;     // байтовый буфер данных
end_var
var_output
    xComplete: bool;            // если = TRUE, то обмен завершен
    Exception: EnModbusException;   // исключения протокола MODBUS или 0xFF – таймаут
    ByteCnt: byte;              // размер прочитанного блока данных, байт
end_var
```

7.5.3.2. Комментарии

Этот ФБ используется для чтения значений аналоговых выходов (AO). В запросе PDU задается начальный адрес первого регистра AO и последующее количество необходимых значений AO. В PDU значения AO адресуются, начиная с нуля. Значение одного регистра в ответе упаковывается в два байта с сетевым порядком байт (big-endian – первый байт содержит старшие биты, второй – младшие). Переменная ByteCnt указывает количество полных байтов данных в ответе, равное удвоенному количеству запрашиваемых значений регистров.

7.5.3.3. Пример

Демонстрационные примеры:

- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExASCIIMasterAsync.project;
- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExRTUMasterAsync.project;
- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExTCPMasterAsync.project.

7.5.4. Функциональный блок ReadInputRegsAsync

Функция 04_{10} (04_{16}) – Read Input Registers (асинхронная).

Пространство имён: AgavaModbusEx.

7.5.4.1. Определение

```
function_block ReadInputRegsAsync
var_input
    xEnable: bool;              // разрешение работы блока
    Mode: EnModbusMode;         // протокол Modbus
    SlaveId: byte (1..247);     // адрес подчинённого устройства в сети MODBUS
    StartAddr: word;            // начальный адрес регистра
    Quantity: byte (1..125);    // количество регистров для чтения
    Handle: RTS_IEC_HANDLE;     // дескриптор соединения
    TimeOut: time;              // время тайм-аута [мс] – макс. задержка на обработку
запроса
    pData: pointer to byte;     // байтовый буфер данных
end_var
var_output
    xComplete: bool;            // если = TRUE, то обмен завершен
    Exception: EnModbusException;   // исключения протокола MODBUS или 0xFF – таймаут
    ByteCnt: byte;              // размер прочитанного блока данных, байт
end_var
```

7.5.4.2. Комментарии

Этот ФБ используется для чтения значений аналоговых входов (AI). В запросе PDU задается начальный адрес первого регистра AI и последующее количество необходимых значений AI. В PDU значения AI адресуются, начиная с нуля. Значение одного регистра в ответе упаковывается в два байта с сетевым порядком байт (big-endian – первый байт содержит старшие биты, второй – младшие). Переменная ByteCnt указывает количество полных байтов данных в ответе, равное удвоенному количеству запрашиваемых значений регистров.

7.5.4.3. Пример

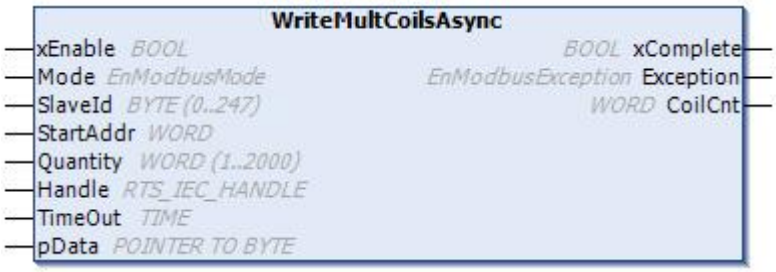
Демонстрационные примеры:

- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExASCIIMasterAsync.project;
- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExRTUMasterAsync.project;
- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExTCPMasterAsync.project.

7.5.5. Функциональный блок WriteSingleCoilAsync

Функция 05_{10} (05_{16}) – Write Single Coil (асинхронная).

Пространство имён: AgavaModbusEx.



7.5.5.1. Определение

```
function_block WriteSingleCoilAsync
var_input
    xEnable: bool;              // разрешение работы блока
    Mode: EnModbusMode;         // протокол Modbus
    SlaveId: byte (0..247);     // адрес подчинённого устройства в сети MODBUS
    CoilAddr: word;             // адрес дискретного выхода / ячейки
    Value: bool;                // значение единичного выхода / ячейки
    Handle: RTS_IEC_HANDLE;     // дескриптор соединения
    TimeOut: time;              // время тайм-аута [мс] – макс. задержка на обработку
запроса
end_var
var_output
    xComplete: bool;            // если = TRUE, то обмен завершен
    Exception: EnModbusException;   // исключения протокола MODBUS или 0xFF – таймаут
end_var
```

7.5.5.2. Комментарии

Этот ФБ используется для записи значения (Value) одного дискретного выхода (DO).

7.5.5.3. Пример

Демонстрационные примеры:

- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExASCIIMasterAsync.project;
- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExRTUMasterAsync.project;
- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExTCPMasterAsync.project.

7.5.6. Функциональный блок WriteSingleRegAsync

Функция 06_{10} (06_{16}) – Write Single Register (асинхронная).

Пространство имён: AgavaModbusEx.

7.5.6.1. Определение

```
function_block WriteSingleRegAsync
var_input
    xEnable: bool;              // разрешение работы блока
    Mode: EnModbusMode;         // протокол Modbus
    SlaveId: byte (0..247);     // адрес подчинённого устройства в сети MODBUS
    RegAddr: word;              // адрес регистра
    Value: word;                // значение регистра
    Handle: RTS_IEC_HANDLE;     // дескриптор соединения
    TimeOut: time;              // время тайм-аута [мс] – макс. задержка на обработку
запроса
end_var
var_output
    xComplete: bool;            // если = TRUE, то обмен завершен
    Exception: EnModbusException;   // исключения протокола MODBUS или 0xFF – таймаут
end_var
```

7.5.6.2. Комментарии

Этот ФБ используется для записи значения (Value) одного аналогового выхода (AO).

7.5.6.3. Пример

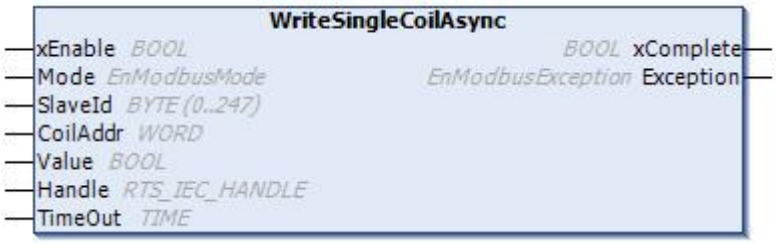
Демонстрационные примеры:

- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExASCIIMasterAsync.project;
- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExRTUMasterAsync.project;
- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExTCPMasterAsync.project.

7.5.7. Функциональный блок WriteMultCoilsAsync

Функция 15_{10} ($0F_{16}$) – Write Single Register (асинхронная).

Пространство имён: AgavaModbusEx.



7.5.7.1. Определение

```
function_block WriteMultCoilsAsync
var_input
    xEnable: bool;              // разрешение работы блока
    Mode: EnModbusMode;         // протокол Modbus
    SlaveId: byte (0..247);     // адрес подчинённого устройства в сети MODBUS
    StartAddr: word;            // начальный адрес дискретных выходов
    Quantity: word (1..2000);   // количество выходов для записи
    Handle: RTS_IEC_HANDLE;     // дескриптор соединения
    TimeOut: time;              // время тайм-аута [мс] – макс. задержка на обработку
запроса
    pData: pointer to byte;     // байтовый буфер данных
end_var
var_output
    xComplete: bool;            // если = TRUE, то обмен завершен
    Exception: EnModbusException;   // исключения протокола MODBUS или 0xFF – таймаут
    CoilCnt: word;              // количество записанных ячеек
end_var
```

7.5.7.2. Комментарии

Этот ФБ используется для записи значений последовательности дискретных выходов (DO). Буфер данных должен содержать упакованные в байты значения DO. Логическая ‘1’ соответствует состоянию ON, ‘0’ – состоянию OFF.

7.5.7.3. Пример

Демонстрационные примеры:

- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExASCIIMasterAsync.project;
- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExRTUMasterAsync.project;
- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExTCPMasterAsync.project.

7.5.8. Функциональный блок WriteMultRegsAsync

Функция 16_{10} (10_{16}) – Write Multiple Registers (асинхронная).

Пространство имён: AgavaModbusEx.

7.5.8.1. Определение

```
function_block WriteMultRegsAsync
var_input
    xEnable: bool;              // разрешение работы блока
    Mode: EnModbusMode;         // протокол Modbus
    SlaveId: byte (0..247);     // адрес подчинённого устройства в сети MODBUS
    StartAddr: word;            // адрес первого регистра
    Quantity: byte (1..125);    // количество регистров для записи
    Handle: RTS_IEC_HANDLE;     // дескриптор соединения
    TimeOut: time;              // время тайм-аута [мс] – макс. задержка на обработку
запроса
    pData: pointer to byte;     // указатель на байтовый буфер данных
end_var
var_output
    xComplete: bool;            // если = TRUE, то обмен завершен
    Exception: EnModbusException;   // исключения протокола MODBUS или 0xFF – таймаут
    RegCnt: byte;               // количество записанных регистров
end_var
```

7.5.8.2. Комментарии

Этот ФБ используется для записи значений последовательности аналоговых выходов (AO). Буфер данных должен содержать значения регистров в формате word с сетевым порядком байт (big-endian – сначала старший байт, затем младший).

7.5.8.3. Пример

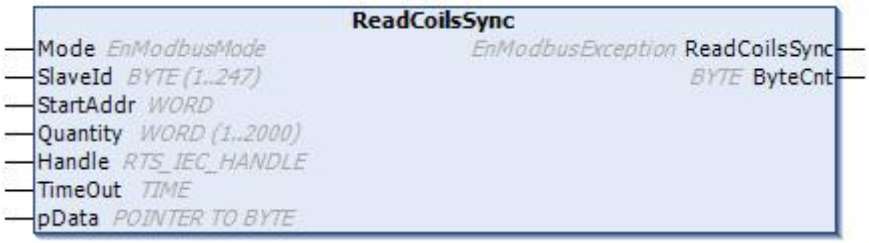
Демонстрационные примеры:

- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExASCIIMasterAsync.project;
- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExRTUMasterAsync.project;
- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExTCPMasterAsync.project.

7.5.9. Функция ReadCoilsSync

Функция 01_{10} (01_{16}) – Read Coils (синхронная).

Пространство имён: AgavaModbusEx.



7.5.9.1. Определение

```
function ReadCoilsSync : EnModbusException
var_input
    Mode: EnModbusMode;          // протокол Modbus
    SlaveId: byte (1..247);      // адрес подчинённого устройства в сети MODBUS
    StartAddr: word;             // начальный адрес дискретных выходов
    Quantity: word (1..2000);    // количество выходов для чтения
    Handle: RTS_IEC_HANDLE;      // дескриптор соединения
    TimeOut: time;               // время тайм-аута [мс] – макс. задержка на обработку
запроса
    pData: pointer to byte;      // байтовый буфер данных
end_var
var_output
    ByteCnt: byte;               // размер прочитанного блока данных, байт
end_var
```

7.5.9.2. Комментарии

Функция используется для чтения значений дискретных выходов (DO). В запросе PDU задается начальный адрес первого регистра DO и последующее количество необходимых значений DO. В PDU значения DO адресуются, начиная с нуля. Значения DO в ответе находятся в одном байте и соответствуют значению битов. Значения битов определяются как 1 = ON и 0 = OFF. Младший бит первого байта данных содержит значение DO, адрес которого указывался в запросе. Остальные значения DO следуют по нарастающей к старшему значению байта. Если запрашиваемое количество не укладывается в целое число байт, то оставшиеся биты в ответе будут заполнены нулями (в направлении от младшего к старшему байту). Переменная ByteCnt указывает количество полных байтов данных в ответе.

В синхронном режиме работы функция не возвращает управление, пока не получит ответ либо пока не закончится время его ожидания.

7.5.9.3. Пример

Демонстрационные примеры:

- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExASCIIMasterSync.project;
- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExRTUMasterSync.project;

- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExTCPMasterSync.project.

7.5.10. Функция ReadDiscrInputsSync

Функция 02_{10} (02_{16}) – Read Discrete Inputs (синхронная).

Пространство имён: AgavaModbusEx.

7.5.10.1. Определение

```
function ReadDiscrInputsSync : EnModbusException
var_input
    Mode: EnModbusMode;          // протокол Modbus
    SlaveId: byte (1..247);      // адрес подчинённого устройства в сети MODBUS
    StartAddr: word;             // начальный адрес дискретных входов
    Quantity: word (1..2000);    // количество входов для чтения
    Handle: RTS_IEC_HANDLE;      // дескриптор соединения
    TimeOut: time;               // время тайм-аута [мс] – макс. задержка на обработку
запроса
    pData: pointer to byte;      // байтовый буфер данных
end_var
var_output
    ByteCnt: byte;               // размер прочитанного блока данных, байт
end_var
```

7.5.10.2. Комментарии

Функция используется для чтения значений дискретных входов (DI). В запросе PDU задается начальный адрес первого регистра DI и последующее количество необходимых значений DI. В PDU значения DI адресуются, начиная с нуля. Значения DI в ответе находятся в одном байте и соответствуют значению битов. Значения битов определяются как 1 = ON и 0 = OFF. Младший бит первого байта данных содержит значение DI, адрес которого указывался в запросе. Остальные значения DI следуют по нарастающей к старшему значению байта. Если запрашиваемое количество не укладывается в целое число байт, то оставшиеся биты в ответе будут заполнены нулями (в направлении от младшего к старшему байту). Переменная ByteCnt указывает количество полных байтов данных в ответе.

В синхронном режиме работы функция не возвращает управление, пока не получит ответ либо пока не закончится время его ожидания.

7.5.10.3. Пример

Демонстрационные примеры:

- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExASCIIMasterSync.project;
- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExRTUMasterSync.project;
- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExTCPMasterSync.project.

7.5.11. Функция ReadHoldRegsSync

Функция 03_{10} (03_{16}) – Read Holding Registers (синхронная).

Пространство имён: AgavaModbusEx.

7.5.11.1. Определение

```
function ReadHoldRegsSync : EnModbusException
var_input
    Mode: EnModbusMode;          // протокол Modbus
    SlaveId: byte (1..247);      // адрес подчинённого устройства в сети MODBUS
    StartAddr: word;             // начальный адрес регистра
    Quantity: byte (1..125);     // количество регистров для чтения
    Handle: RTS_IEC_HANDLE;      // дескриптор соединения
    TimeOut: time;               // время тайм-аута [мс] – макс. задержка на обработку
запроса
    pData: pointer to byte;      // байтовый буфер данных
end_var
var_output
    ByteCnt: byte;               // размер прочитанного блока данных, байт
end_var
```

7.5.11.2. Комментарии

Функция используется для чтения значений аналоговых выходов (AO). В запросе PDU задается начальный адрес первого регистра AO и последующее количество необходимых значений AO. В PDU значения AO адресуются, начиная с нуля. Значение одного регистра в ответе упаковывается в два байта с сетевым порядком байт (big-endian – первый байт содержит старшие биты, второй – младшие). Переменная ByteCnt указывает количество полных байтов данных в ответе, равное удвоенному количеству запрашиваемых значений регистров.

В синхронном режиме работы функция не возвращает управление, пока не получит ответ либо пока не закончится время его ожидания.

7.5.11.3. Пример

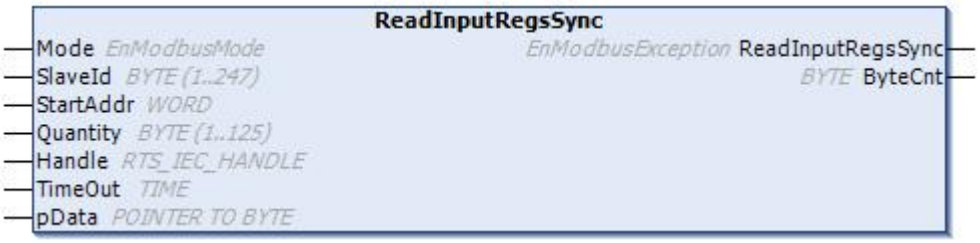
Демонстрационные примеры:

- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExASCIIMasterSync.project;
- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExRTUMasterSync.project;
- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExTCPMasterSync.project.

7.5.12. Функция ReadInputRegsSync

Функция 04_{10} (04_{16}) – Read Input Registers (синхронная).

Пространство имён: AgavaModbusEx.



7.5.12.1. Определение

```
function ReadInputRegsSync : EnModbusException
var_input
    Mode: EnModbusMode;          // протокол Modbus
    SlaveId: byte (1..247);      // адрес подчинённого устройства в сети MODBUS
    StartAddr: word;             // начальный адрес регистра
    Quantity: byte (1..125);     // количество регистров для чтения
    Handle: RTS_IEC_HANDLE;      // дескриптор соединения
    TimeOut: time;               // время тайм-аута [мс] – макс. задержка на обработку
запроса
    pData: pointer to byte;      // указатель на байтовый буфер данных
end_var
var_output
    ByteCnt: byte;               // размер прочитанного блока данных, байт
end_var
```

7.5.12.2. Комментарии

Функция используется для чтения значений аналоговых входов (AI). В запросе PDU задается начальный адрес первого регистра AI и последующее количество необходимых значений AI. В PDU значения AI адресуются, начиная с нуля. Значение одного регистра в ответе упаковывается в два байта с сетевым порядком байт (big-endian – первый байт содержит старшие биты, второй – младшие). Переменная ByteCnt указывает количество полных байтов данных в ответе, равное удвоенному количеству запрашиваемых значений регистров.

В синхронном режиме работы функция не возвращает управление, пока не получит ответ либо пока не закончится время его ожидания.

7.5.12.3. Пример

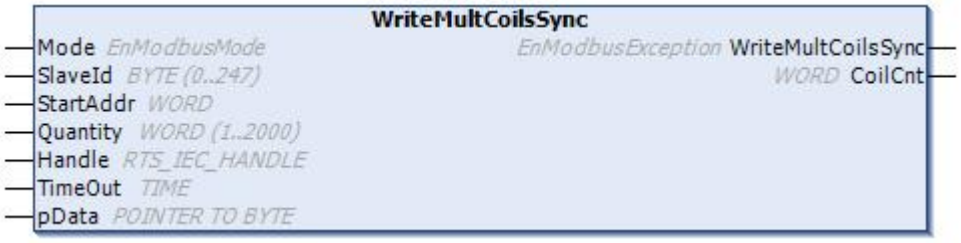
Демонстрационные примеры:

- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExASCIIMasterSync.project;
- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExRTUMasterSync.project;
- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExTCPMasterSync.project.

7.5.13. Функция WriteSingleCoilSync

Функция 05_{10} (05_{16}) – Write Single Coil (синхронная).

Пространство имён: AgavaModbusEx.



7.5.13.1. Определение

```
function WriteSingleCoilSync : EnModbusException
var_input
    Mode: EnModbusMode;          // протокол Modbus
    SlaveId: byte (0..247);      // адрес подчинённого устройства в сети MODBUS
    CoilAddr: word;              // адрес дискретного выхода / ячейки
    Value: bool;                 // значение единичного выхода / ячейки
    Handle: RTS_IEC_HANDLE;      // дескриптор соединения
    TimeOut: time;               // время тайм-аута [мс] – макс. задержка на обработку
запроса
end_var
```

7.5.13.2. Комментарии

Функция используется для записи значения (Value) одного дискретного выхода (DO).

В синхронном режиме работы функция не возвращает управление, пока не получит ответ либо пока не закончится время его ожидания.

7.5.13.3. Пример

Демонстрационные примеры:

- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExASCIIMasterSync.project;
- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExRTUMasterSync.project;
- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExTCPMasterSync.project.

7.5.14. Функция WriteSingleRegSync

Функция 06_{10} (06_{16}) – Write Single Register (синхронная).

Пространство имён: AgavaModbusEx.

7.5.14.1. Определение

```
function WriteSingleRegSync : EnModbusException
var_input
    Mode: EnModbusMode;          // протокол Modbus
    SlaveId: byte (0..247);      // адрес подчинённого устройства в сети MODBUS
    RegAddr: word;               // адрес регистра
    Value: word;                 // значение регистра
    Handle: RTS_IEC_HANDLE;      // дескриптор соединения
    TimeOut: time;               // время тайм-аута [мс] – макс. задержка на обработку
запроса
end_var
```

7.5.14.2. Комментарии

Функция используется для записи значения (Value) одного аналогового выхода (AO).

В синхронном режиме работы функция не возвращает управление, пока не получит ответ либо пока не закончится время его ожидания.

7.5.14.3. Пример

Демонстрационные примеры:

- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExASCIIMasterSync.project;
- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExRTUMasterSync.project;
- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExTCPMasterSync.project.

7.5.15. Функция WriteMultCoilsAsync

Функция 15_{10} ($0F_{16}$) – Write Single Register (синхронная).

Пространство имён: AgavaModbusEx.

7.5.15.1. Определение

```
function WriteMultCoilsSync : EnModbusException
var_input
    Mode: EnModbusMode;          // протокол Modbus
    SlaveId: byte (0..247);      // адрес подчинённого устройства в сети MODBUS
    StartAddr: word;             // начальный адрес дискретных выходов
    Quantity: word (1..2000);    // количество выходов для записи
    Handle: RTS_IEC_HANDLE;      // дескриптор соединения
    TimeOut: time;               // время тайм-аута [мс] – макс. задержка на обработку
запроса
    pData: pointer to byte;      // указатель на байтовый буфер данных
end_var
var_output
    CoilCnt: word;               // количество записанных ячеек
end_var
```

7.5.15.2. Комментарии

Функция используется для записи значений последовательности дискретных выходов (DO). Буфер данных должен содержать упакованные в байты значения DO. Логическая ‘1’ соответствует состоянию ON, ‘0’ – состоянию OFF.

В синхронном режиме работы функция не возвращает управление, пока не получит ответ либо пока не закончится время его ожидания.

7.5.15.3. Пример

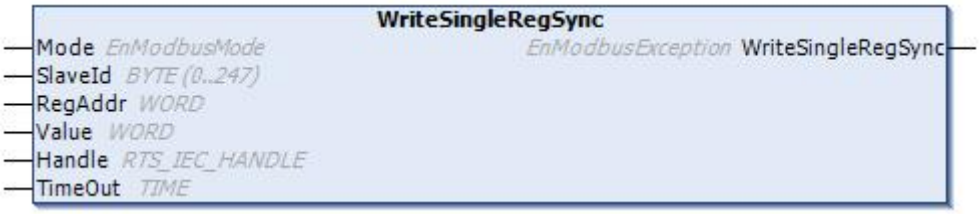
Демонстрационные примеры:

- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExASCIIMasterSync.project;
- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExRTUMasterSync.project;
- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExTCPMasterSync.project.

7.5.16. Функция WriteMultRegsSync

Функция 16_{10} (10_{16}) – Write Multiple Registers (синхронная).

Пространство имён: AgavaModbusEx.



7.5.16.1. Определение

```
function WriteMultRegsSync : EnModbusException
var_input
    Mode: EnModbusMode;          // протокол Modbus
    SlaveId: byte (0..247);      // адрес подчинённого устройства в сети MODBUS
    StartAddr: word;             // адрес первого регистра
    Quantity: byte (1..125);     // количество регистров для записи
    Handle: RTS_IEC_HANDLE;      // дескриптор соединения
    TimeOut: time;               // время тайм-аута [мс] – макс. задержка на обработку
запроса
    pData: pointer to byte;      // указатель на байтовый буфер данных
end_var
var_output
    RegCnt: byte;                // количество записанных регистров
end_var
```

7.5.16.2. Комментарии

Функция используется для записи значений последовательности аналоговых выходов (AO). Буфер данных должен содержать значения регистров в формате word с сетевым порядком байт (big-endian – сначала старший байт, затем младший).

В синхронном режиме работы функция не возвращает управление, пока не получит ответ либо пока не закончится время его ожидания.

7.5.16.3. Пример

Демонстрационные примеры:

- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExASCIIMasterSync.project;
- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExRTUMasterSync.project;
- Примеры\AgavaModbusEx\ПЛК40\ST\ModbusExTCPMasterSync.project.

7.5.17. Перечисления

Пространство имён: AgavaModbus.

7.5.17.1. EnModbusException

Основные ошибки обмена по Modbus.

7.5.17.1.1. Определение

```
{attribute 'qualified_only'}
type EnModbusException:
(
    Success := 0,           // успешно
    TimeoutError := -1,     // время ожидания истекло
    HardwareError := -2,    // ошибка соединения
    InputDataError := -3    // ошибка данных
);
end_type
```

7.5.17.2. EnModbusMode

Режимы работы по протоколу Modbus.

7.5.17.2.1. Определение

```
{attribute 'qualified_only'}
type EnModbusMode:
(
    RTU := 0,   // RTU (Remote Terminal Unit)
    ASCII,      // ASCII
    TCP         // TCP
);
end_type
```

7.6. Библиотека OwenModules

Библиотека версии 3.5.10.1 содержит функциональный блок, предназначенный для работы с модулями ввода-вывода фирмы Owen.

Библиотека поставляется с исходными текстами. Пользователь может расширять функционал библиотеки по аналогии с имеющимися примерами работы с модулями. Расширение предполагает добавление структуры данных, соответствующую типу модуля, и дополнение функционального блока группой методов, осуществляющих взаимодействие с модулем.

Функциональный блок поддерживает работу только по протоколу Modbus RTU и использует для этого синхронные функции обмена из библиотеки AgavaModulesEx.

7.6.1. Функциональный блок TExtIoMxSync

Пространство имён: OwenModules.

7.6.1.1. Определение

```
// Функциональный блок внешнего модуля ввода-вывода фирмы Овен.
function_block TExtIoMxSync
```

7.6.1.2. Входы

Название	Тип	Описание
xEnable	bool	Разрешение работы блока
handle	RTS_IEC_HANDLE	Дескриптор линии связи
devid	byte	Адрес устройства (для последовательной линии)
timeout	time	Время ожидания обработки запроса
pMxData	pointer to byte	Указатель на буфер данных модуля

7.6.1.3. Выходы

Название	Тип	Описание
xError	bool	Состояние ошибки
resIEC	RTS_IEC_RESULT	Результат выполнения: ERR_OK - успешно
nSend	dword	Число запросов модуля
nRsv	dword	Число ответов от модуля

7.6.1.4. Комментарии

Функциональный блок предоставляет универсальный доступ к данным модулей ввода-вывода серии Mx110.

7.6.1.5. Примеры

Демонстрационные примеры на языке CFC:

Примеры\OwenModules\ПЛК40\CFC\MU110_8R_K.project,
Примеры\OwenModules\ПЛК40\CFC\MV110_8AS.project,
Примеры\OwenModules\ПЛК40\CFC\MV110_16D_DN.project.

7.6.2. Структуры

Пространство имён: OwenModules.

7.6.2.1. TMU110_8R_K

Данные модуля типа МУ110-8Р (К).

7.6.2.1.1. Определение

```
{attribute 'pack_mode' := '0'}
type TMU110_8R_K:
struct
    mxType: EnMxType := EnMxType.MU110_8R_K;  // Тип модуля ввода-вывода
    xPWM: bool;     // Тип управления: false - дискретный, true - ШИМ
    Qerrpwm1: uint; // Скважность ШИМ выхода 1 (при аварии), [0.1 %]
    Qerrpwm2: uint; // Скважность ШИМ выхода 2 (при аварии), [0.1 %]
    Qerrpwm3: uint; // Скважность ШИМ выхода 3 (при аварии), [0.1 %]
    Qerrpwm4: uint; // Скважность ШИМ выхода 4 (при аварии), [0.1 %]
    Qerrpwm5: uint; // Скважность ШИМ выхода 5 (при аварии), [0.1 %]
    Qerrpwm6: uint; // Скважность ШИМ выхода 6 (при аварии), [0.1 %]
    Qerrpwm7: uint; // Скважность ШИМ выхода 7 (при аварии), [0.1 %]
    Qerrpwm8: uint; // Скважность ШИМ выхода 8 (при аварии), [0.1 %]
    Tpwm1: uint;    // Период ШИМ выхода 1, [сек]
    Tpwm2: uint;    // Период ШИМ выхода 2, [сек]
    Tpwm3: uint;    // Период ШИМ выхода 3, [сек]
    Tpwm4: uint;    // Период ШИМ выхода 4, [сек]
    Tpwm5: uint;    // Период ШИМ выхода 5, [сек]
    Tpwm6: uint;    // Период ШИМ выхода 6, [сек]
    Tpwm7: uint;    // Период ШИМ выхода 7, [сек]
    Tpwm8: uint;    // Период ШИМ выхода 8, [сек]
    Qpwm1: uint;    // Скважность ШИМ выхода 1, [0.1 %]
    Qpwm2: uint;    // Скважность ШИМ выхода 2, [0.1 %]
    Qpwm3: uint;    // Скважность ШИМ выхода 3, [0.1 %]
    Qpwm4: uint;    // Скважность ШИМ выхода 4, [0.1 %]
    Qpwm5: uint;    // Скважность ШИМ выхода 5, [0.1 %]
    Qpwm6: uint;    // Скважность ШИМ выхода 6, [0.1 %]
    Qpwm7: uint;    // Скважность ШИМ выхода 7, [0.1 %]
    Qpwm8: uint;    // Скважность ШИМ выхода 8, [0.1 %]
    wOut1: bool;    // Состояние выхода 1 (запись)
    wOut2: bool;    // Состояние выхода 2 (запись)
    wOut3: bool;    // Состояние выхода 3 (запись)
    wOut4: bool;    // Состояние выхода 4 (запись)
    wOut5: bool;    // Состояние выхода 5 (запись)
    wOut6: bool;    // Состояние выхода 6 (запись)
    wOut7: bool;    // Состояние выхода 7 (запись)
    wOut8: bool;    // Состояние выхода 8 (запись)
    rOut1: bool;    // Состояние выхода 1 (чтение)
    rOut2: bool;    // Состояние выхода 2 (чтение)
    rOut3: bool;    // Состояние выхода 3 (чтение)
    rOut4: bool;    // Состояние выхода 4 (чтение)
    rOut5: bool;    // Состояние выхода 5 (чтение)
    rOut6: bool;    // Состояние выхода 6 (чтение)
    rOut7: bool;    // Состояние выхода 7 (чтение)
    rOut8: bool;    // Состояние выхода 8 (чтение)
```

```
end_struct
end_type
```

7.6.2.2. TMV110_16D_DN

Данные модуля типа МВ110-16Д (ДН).

7.6.2.2.1. Определение

```
{attribute 'pack_mode' := '0'}
type TMV110_16D_DN:
struct
    mxType: EnMxType := EnMxType.MV110_16D_DN;  // Тип модуля ввода-вывода
    wCounter1: word;    // Значение счетчика входа 1 (запись)
    wCounter2: word;    // Значение счетчика входа 2 (запись)
    wCounter3: word;    // Значение счетчика входа 3 (запись)
    wCounter4: word;    // Значение счетчика входа 4 (запись)
    wCounter5: word;    // Значение счетчика входа 5 (запись)
    wCounter6: word;    // Значение счетчика входа 6 (запись)
    wCounter7: word;    // Значение счетчика входа 7 (запись)
    wCounter8: word;    // Значение счетчика входа 8 (запись)
    wCounter9: word;    // Значение счетчика входа 9 (запись)
    wCounter10: word;   // Значение счетчика входа 10 (запись)
    wCounter11: word;   // Значение счетчика входа 11 (запись)
    wCounter12: word;   // Значение счетчика входа 12 (запись)
    wCounter13: word;   // Значение счетчика входа 13 (запись)
    wCounter14: word;   // Значение счетчика входа 14 (запись)
    wCounter15: word;   // Значение счетчика входа 15 (запись)
    wCounter16: word;   // Значение счетчика входа 16 (запись)
    rInput1: bool;      // Состояние входа 1 (чтение)
    rInput2: bool;      // Состояние входа 2 (чтение)
    rInput3: bool;      // Состояние входа 3 (чтение)
    rInput4: bool;      // Состояние входа 4 (чтение)
    rInput5: bool;      // Состояние входа 5 (чтение)
    rInput6: bool;      // Состояние входа 6 (чтение)
    rInput7: bool;      // Состояние входа 7 (чтение)
    rInput8: bool;      // Состояние входа 8 (чтение)
    rInput9: bool;      // Состояние входа 9 (чтение)
    rInput10: bool;     // Состояние входа 10 (чтение)
    rInput11: bool;     // Состояние входа 11 (чтение)
    rInput12: bool;     // Состояние входа 12 (чтение)
    rInput13: bool;     // Состояние входа 13 (чтение)
    rInput14: bool;     // Состояние входа 14 (чтение)
    rInput15: bool;     // Состояние входа 15 (чтение)
    rInput16: bool;     // Состояние входа 16 (чтение)
    rCounter1: word;    // Счетчик входа 1 (чтение)
    rCounter2: word;    // Счетчик входа 2 (чтение)
    rCounter3: word;    // Счетчик входа 3 (чтение)
    rCounter4: word;    // Счетчик входа 4 (чтение)
    rCounter5: word;    // Счетчик входа 5 (чтение)
    rCounter6: word;    // Счетчик входа 6 (чтение)
    rCounter7: word;    // Счетчик входа 7 (чтение)
    rCounter8: word;    // Счетчик входа 8 (чтение)
    rCounter9: word;    // Счетчик входа 9 (чтение)
    rCounter10: word;   // Счетчик входа 10 (чтение)
    rCounter11: word;   // Счетчик входа 11 (чтение)
```

```
    rCounter12: word;   // Счетчик входа 12 (чтение)
    rCounter13: word;   // Счетчик входа 13 (чтение)
    rCounter14: word;   // Счетчик входа 14 (чтение)
    rCounter15: word;   // Счетчик входа 15 (чтение)
    rCounter16: word;   // Счетчик входа 16 (чтение)
end_struct
end_type
```

7.6.2.3. TMV110_8AS

Данные модуля типа MB110-8AC.

7.6.2.3.1. Определение

```
{attribute 'pack_mode' := '0'}
type TMV110_8AS:
struct
    mxType: EnMxType := EnMxType.MV110_8AS;   // Тип модуля ввода-вывода
    InType_1: word := 1;    // Тип подключаемого датчика. Канал 1
    InType_2: word := 1;    // Тип подключаемого датчика. Канал 2
    InType_3: word := 1;    // Тип подключаемого датчика. Канал 3
    InType_4: word := 1;    // Тип подключаемого датчика. Канал 4
    InType_5: word := 1;    // Тип подключаемого датчика. Канал 5
    InType_6: word := 1;    // Тип подключаемого датчика. Канал 6
    InType_7: word := 1;    // Тип подключаемого датчика. Канал 7
    InType_8: word := 1;    // Тип подключаемого датчика. Канал 8
    SRD_1: word;            // Статус измерения в канале 1 (код ошибки)
    SRD_2: word;            // Статус измерения в канале 2 (код ошибки)
    SRD_3: word;            // Статус измерения в канале 3 (код ошибки)
    SRD_4: word;            // Статус измерения в канале 4 (код ошибки)
    SRD_5: word;            // Статус измерения в канале 5 (код ошибки)
    SRD_6: word;            // Статус измерения в канале 6 (код ошибки)
    SRD_7: word;            // Статус измерения в канале 7 (код ошибки)
    SRD_8: word;            // Статус измерения в канале 8 (код ошибки)
    Read_1: real;           // Показания канала 1 в представлении с плавающей точкой
    C_Time_1: word;         // Метка относительно времени. Канал 1
    Read_2: real;           // Показания канала 1 в представлении с плавающей точкой
    C_Time_2: word;         // Метка относительно времени. Канал 2
    Read_3: real;           // Показания канала 1 в представлении с плавающей точкой
    C_Time_3: word;         // Метка относительно времени. Канал 3
    Read_4: real;           // Показания канала 1 в представлении с плавающей точкой
    C_Time_4: word;         // Метка относительно времени. Канал 4
    Read_5: real;           // Показания канала 1 в представлении с плавающей точкой
    C_Time_5: word;         // Метка относительно времени. Канал 5
    Read_6: real;           // Показания канала 1 в представлении с плавающей точкой
    C_Time_6: word;         // Метка относительно времени. Канал 6
    Read_7: real;           // Показания канала 1 в представлении с плавающей точкой
    C_Time_7: word;         // Метка относительно времени. Канал 7
    Read_8: real;           // Показания канала 1 в представлении с плавающей точкой
    C_Time_8: word;         // Метка относительно времени. Канал 8
end_struct
end_type
```

7.6.3. Перечисления

Пространство имён: OwenModules.

7.6.3.1. EnMxType

Типы модулей ввода/вывода фирмы Овен.

7.6.3.1.1. Определение

```
{attribute 'qualified_only'}
type EnMxType:
(
    Unknown         := 0,
    MV110_8AS       := 103,
    MV110_16D_DN    := 104,
    MU110_8R_K      := 111
);
end_type
```

8. Список рекомендуемой литературы