



## Контроллеры промышленные серии MX300

Руководство по эксплуатации

Редакция от августа 2024 года

[optimusdrive.ru](http://optimusdrive.ru)

## Оглавление

Введение.....	4
Меры предосторожности при эксплуатации.....	5
Функциональное назначение .....	6
Перечень оборудования .....	7
Спецификация ЦПУ (контроллеров).....	8
Спецификация дискретных входов-выходов на ЦПУ .....	9
Внешний вид и размеры ЦПУ .....	10
Расположение клемм ЦПУ .....	11
Схемы подключения МХ308-СЕ/МХ316-СЕ/МХ332-СЕ .....	12
Расположение интерфейсов МХ308-СЕ/МХ316-СЕ/МХ332-СЕ.....	16
Спецификация источника питания .....	20
Установка модулей расширения.....	21
Спецификация модулей дискретных входов-выходов .....	23
Внешний вид и размеры дискретных модулей расширения.....	24
Расположение клемм дискретных модулей расширения.....	25
Схемы подключения дискретных входов-выходов модулей расширения.....	29
Спецификация модулей аналоговых входов-выходов .....	33
Внешний вид и размеры аналоговых модулей .....	35
Расположение клемм аналоговых модулей .....	36
Схемы подключения аналоговых входов-выходов .....	37
Запуск среды программирования и создание проекта .....	39
Установка описания устройства для контроллера МХ300.....	40
Определение версии библиотеки 3S SoftMotion (SM3) .....	42
Добавление контроллера в проект.....	43
Организация связи контроллеров типа МХ300 и среды программирования. Загрузка программы. Онлайн режим .....	44
Использование встроенных входов-выходов контроллера в обычном режиме.....	60
Добавление в проект модулей расширения .....	62
Поддерживаемые базовые типы данных .....	74
Список наиболее употребительных команд .....	75
Добавление в проект сервопривода .....	87
Работа с высокоскоростными счётчиками .....	100
Импульсная ось движения .....	107
Применение штурвального энкодера в качестве мастер-оси .....	113
Последовательная связь по протоколу Modbus RTU Master.....	115
Последовательная связь по протоколу Modbus RTU Slave.....	123
Связь по протоколу Modbus TCP Master .....	128
Связь по протоколу Modbus TCP Slave .....	137
Работа с преобразователем частоты серии Optimus Drive AD800 по сети EtherCAT..	144
Связь по протоколу Ethernet/IP Scanner (Master).....	153

---

Связь по протоколу Ethernet/IP Adapter (Slave).....	172
Чтение и установка часов реального времени.....	184

[www.optimusdrive.ru](http://www.optimusdrive.ru)

## Введение

Настоящее Руководство описывает порядок и особенности использования контроллеров серии МХ300 в среде программирования DesignerAX 1.5 или CODESYS 3.5.18.30 и содержит информацию по техническим характеристикам контроллеров и модулей расширения, а также основным операциям с ними.

Редакция от марта 2024 года содержит следующие сведения:

- технические характеристики контроллеров и модулей расширения
- установка конфигурационных файлов
- подключение контроллера к ПК
- работа с модулями ввода-вывода

Редакция от июня 2024 года содержит следующие сведения:

- Применение штурвального энкодера в качестве мастер-оси
- Последовательная связь по протоколу Modbus RTU Master
- Последовательная связь по протоколу Modbus RTU Slave
- Связь по протоколу Modbus TCP Master
- Связь по протоколу Modbus TCP Slave
- Работа с преобразователем частоты серии Optimus Drive AD800 по сети EtherCAT

***За информацией по работе в самой среде разработки CODESYS обращайтесь на сайт компании разработчика данного программного обеспечения.***

## Меры предосторожности при эксплуатации

**Контроллеры семейства МХ300 и модули расширения к ним предназначены для использования только квалифицированным персоналом!**

**Перед началом эксплуатации внимательно ознакомьтесь с настоящим Руководством!**

### Меры предосторожности при монтаже и установке

#### ВНИМАНИЕ

- Не наступайте на контроллер (модули расширения) и не кладите на него тяжелые предметы
- Не блокируйте вентиляционные отверстия и не допускайте попадания в них посторонних частиц
- Контроллер и модули можно устанавливать только вертикально с обеспечением свободного пространства не менее 50 мм со всех сторон. В шкафу должна быть обеспечена свободная конвенция воздуха
- Не подвергайте контроллер и модули ударам
- Контроллер и модули имеют степень защиты IP20 и не являются водонепроницаемым. Примите меры, чтобы предотвратить попадание воды и т.п. внутрь контроллера и модулей
- Контроллер и модули предназначены для установки только в общую защитную оболочку (шкаф управления). Эксплуатация в открытом виде запрещена

### Меры предосторожности при подключении и работе

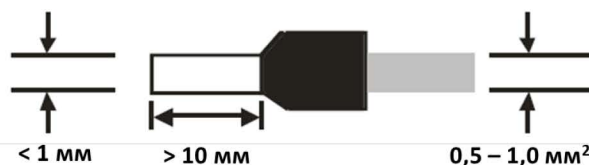
#### ВНИМАНИЕ

- Подключите кабели заземления во избежание поражения электрическим током и пожара, а также должной работе экранов кабелей связи
- Напряжение питания контроллера и модулей строго 24 VDC. Используйте только стабилизированные источники питания достаточной мощности. Учитывайте при выборе источника питания пусковые токи. Категорически запрещается подключение к контроллеру переменного напряжения. Нарушение данного требования однозначно выведет контроллер (модули) из строя
- Проверьте затяжку винтов клемм, неполная затяжка может привести к возгоранию
- Во избежание несчастных случаев и выхода из строя оборудования обратите внимание на правильность подключения кабелей

### Рекомендации по использованию проводов и наконечников

На контроллерах МХ300 и модулях расширения установлены пружинные клеммники, которые позволяют производить монтаж проводов без наконечников. Рекомендуются строго гибкие (многопроволочные) провода сечением 0,5 – 1,0 мм<sup>2</sup>.

В случае использования наконечников рекомендуются наконечники без юбки. Если Вы принципиально используете наконечники с юбкой, то длина рабочей части наконечника должна быть не менее 10 мм. Диаметр после обжима не более 1 мм.



## Функциональное назначение

Приборы семейства MX300 являются многофункциональными контроллерами, предназначенными для решения широкого круга задач как общей автоматизации в промышленности, так и задач управления сложным движением по шине EtherCAT с большой скоростью процессов. Обладают высокими программными возможностями с поддержкой сложных вычислений, логики и операций с данными. Имеют встроенные порты Ethernet, EtherCAT, RS232, RS485, mini USB. Поддерживают протоколы связи EtherCAT, Ethernet/IP, Modbus, OPC UA.

Для программирования контроллеров семейства MX300 рекомендуется среда разработки Designer-AX 1.5 и выше версией, которую можно скачать с сайта [www.optimusdrive.ru](http://www.optimusdrive.ru) Возможно использование среды программирования CODESYS 3.5.18.30 и выше версией. Но в данном случае пользователь контроллера изыскивает её самостоятельно.

Designer-AX предоставляет следующие инструменты для разработки проекта:

- Языки программирования стандарта IEC 61131-3: LD, ST, CFC, SFC и FBD
- Программные объекты типа POU, FB, FC, Interface, DUT, Task и др.
- Поддержка большого количества типов данных
- Большая библиотека прикладных команд для различных применений
- Всплывающие подсказки при вводе и настройке
- Развитый интерфейс программирования и настройки
- Различные инструменты отладки, симулятор, онлайн режим, правка программы в онлайн
- Многоуровневая защита исходного кода проекта
- Поддержка устройств разных производителей

### Функции управления движением:

- CODESYS SM3\_Basics/Robotics/CNC v.4.10
- LS Motion Lib для интерполированного движения,
- динамический E-CAM, EGear, диагностика, мониторинг данных
- Поддержка физических и логических осей;
- Графический редактор E-CAM;

IP адрес по умолчанию: **192.168.1.3**

*USB драйвер и XML файлы* модулей расширения устанавливаются в составе пакета установки.

Система состоит из Центрального Процессорного Устройства (ЦПУ), т.е. контроллера, и модулей расширения дискретных и аналоговых входов-выходов. Далее приводится перечень оборудования, входящего в состав системы на основе контроллеров MX300, а также приводятся характеристики аппаратной части Центральных Процессорных Устройств (ЦПУ). Характеристики модулей расширения приводятся в отдельном Руководстве.

## Перечень оборудования

Наименование	Обозначение	Описание
Контроллеры (ЦПУ)	MX308-CE	Контроллер, EtherCAT 8 осей движения, 6 импульсных осей 200 кГц, 16DI/16DO (NPN), Ethernet, RS232, 2xRS485, CAN, микро SD, USB-C, 24 VDC
	MX316-CE	Контроллер, EtherCAT 8 осей движения, 6 импульсных осей 200 кГц, 16DI/16DO (NPN), Ethernet, RS232, 2xRS485, CAN, микро SD, USB-C, 24 VDC
	MX332-CE	Контроллер, EtherCAT 8 осей движения, 6 импульсных осей 200 кГц, 16DI/16DO (NPN), Ethernet, RS232, 2xRS485, CAN, микро SD, USB-C, 24 VDC
Модули дискретных входов/выходов	MX16DI	24 В DC 5 мА 16 входов NPN/PNP Пружинный клеммный блок
	MX16DOP	5 ~ 30 В DC 0.5 А 16 выходов Выходы: PNP Пружинный клеммный блок
	MX16DOR	240 В AC / 24 В DC 2 А 16 выходов Выходы: Реле Пружинный клеммный блок
	MX16DOT	5 ~ 30VDC 0.5А 16 выходов Выходы: NPN Пружинный клеммный блок
Модули аналоговых входов-выходов	MX04AD	4-канальный модуль аналоговых входов Аппаратное разрешение: 16 бит 0~10 В, 0/1~5 В, -5~+5 В, -10~-+10 В, 0/4~20 мА, -20~-+20 мА Время преобразования: 2 мс/канал
	MX04DA	4-канальный модуль аналоговых выходов Аппаратное разрешение: 12 бит 0~10 В, 0/1~5 В, -5~+5 В, -10~-+10 В, 0/4~20 мА Время преобразования: 2 мс/канал
Станция удалённого ввода-вывода EtherCAT	R2EC	Использует те же модули расширения, что и ЦПУ
Патч-корды EtherCAT	CABLE-TX0M3-BUS	0.3 м
	CABLE-TX0M5-BUS	0.5 м
	CABLE-TX1M0-BUS	1 м
	CABLE-TX2M0-BUS	2 м
	CABLE-TX5M0-BUS	5 м
	CABLE-TX10M0-BUS	10 м
	CABLE-TX20M0-BUS	20 м

**Примечание:** Термин «ось движения» означает, что данный контроллер позволяет осуществлять скоординированное движение, т.е. группировать оси для совместного движения, например линейной и круговой интерполяции, E-CAM, GEAR и т.п.

## Спецификация ЦПУ (контроллеров)

Модель	MX308-CE	MX316-CE	MX332-CE
Кол-во поддерживаемых осей	8 осей EtherCAT + импульсное управление 6 осями 200 кГц	16 осей EtherCAT + импульсное управление 6 осями 200 кГц	16 осей EtherCAT + импульсное управление 6 осями 200 кГц
Процессор	ARM Cortex-A9 Dual Core 866 МГц		
Макс. кол-во модулей расширения	32		
EtherNET	1* EtherNET port, Modbus TCP, Socket, загрузка и выгрузка программы, отладка		
EtherCAT	EtherCAT master, поддержка до 128 ведомых устройств		
Порт последовательной связи	RS232, 2xRS485, пользовательский протокол, MODBUS RTU Ведущий/Ведомый		
Поддержка устройств по шине CANopen	до 31 Ведомого		
Память под программу	20 Мб		
Память данных	40 Мб		
Энергонезависимая память	512 Кбайт		
Порт USB	Тип С, загрузка и выгрузка программы, отладка		
Слот карты памяти	карта micro SD card, FAT32, до 32 Гб, для переноса программы пользователя		
Управление движением	Точка-точка, электронный кулачок (E-CAM), интерполяция		
Высокоскоростные счетчики	6 входов АВ (200 кГц)		
Встроенные входы/выходы	16 входов (12 входов 200 кГц/4 входа 1 кГц(NPN/PNP)) 16 выходов (12 выходов 200 кГц/4 выхода 10 кГц (NPN))		
Часы реального времени	Да (встроенная батарейка CR2032 с выводами и разъемом)		
Среда разработки	Designer-AX 1.5, CODESYS 3.5.18.30		
Языки программирования	ST, LD, CFC, SFC, FBD		
Библиотеки	SM3_Basics/Robotics/CNC v.4.10		
Напряжение питания	24 В постоянного тока		
Потребляемая мощность	3.6 Вт		
Рабочая температура	0~55°C		
Габаритные размеры	ВхШхГ: 100.00 x 81.75 x 98,5 мм		



## Спецификация дискретных входов-выходов на ЦПУ

### Спецификация дискретных входов:

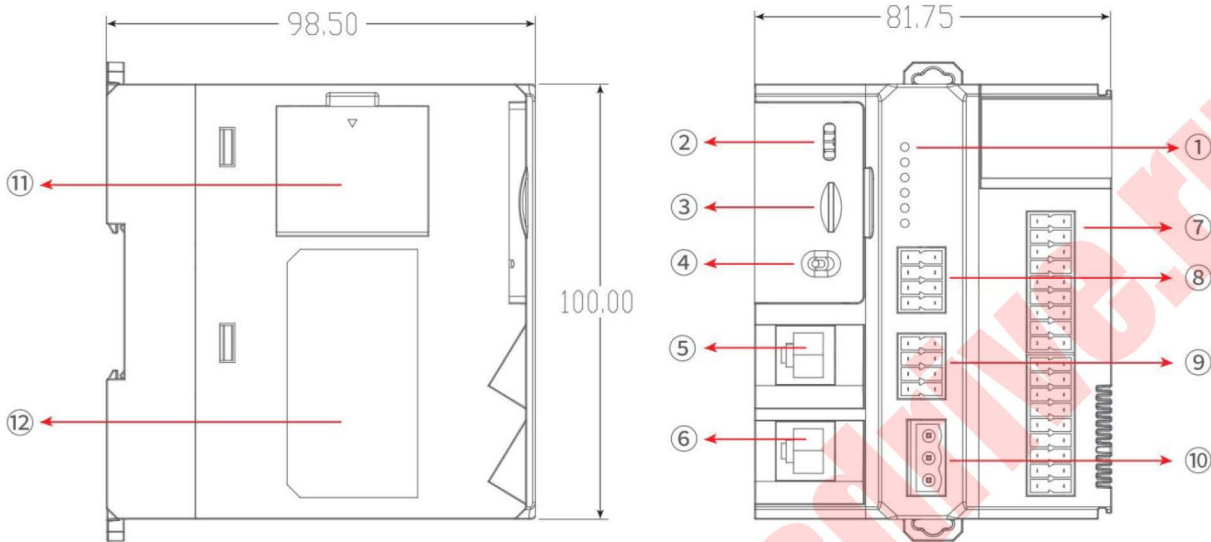
Модель		MX308-CE	MX316-CE	MX332-CE
Количество входов		16		
Тип соединения		Съёмный пружинный клеммник		
Тип входа		Дискретный вход		
Форма входа		Постоянный ток (NPN/PNP, две общие точки SS0/SS1) SS0 для IN0~IN7, SS1 для IN8~IN15		
Напряжение/ ток		24 VDC, 5 mA		
Уровень вкл/выкл	OFF→ON	>15 VDC		
	ON→OFF	<5 VDC		
Максимальная входная частота		200 KHz		
Входное сопротивление		4.7 kΩ обычные входы, 3.3 kΩ импульсные входы		
Тип входного сигнала		Потенциальный сигнал Sinking: SS0/SS1 подключены к 24V+ Sourcing: SS0/SS1 подключены к 0V		
Электрическая изоляция		оптопары		
Индикация		Когда оптопара активна, светодиод входа включен		

### Спецификация дискретных выходов:

Модель		MX308-CE	MX316-CE	MX332-CE
Количество выходов		16		
Тип соединения		Съёмный пружинный клеммник		
Тип выхода		NPN (Sinking)		
Напряжение		5~30 VDC		
Макси- мальная нагрузка	Активная	0.5A/выход		
	Индуктивная	-		
	Лампочка	-		
Максимальная выходная частота		200 KHz		
Общая точка COM		Одна общая точка на все выходы, 2.4A/COM		

## Внешний вид и размеры ЦПУ

MX308-CE/MX316-CE/MX332-CE



### 1 Светодиоды:

POWER: Питание (зеленый), SYS: Индикатор работы (зеленый), RUN: шина EtherCAT (зеленый),  
ERR: ошибка сети EtherCAT (красный)

2 Порт USB Type-C

3 Карта Micro SD

4 Переключатель RUN/STOP

5 Шина EtherNet: OPC UA, EtherNet/IP, MODBUS TCP

6 Шина EtherCAT

7 Встроенные входы/выходы

8 Порты последовательной связи 2xRS485

9 Порты CANopen и RS232

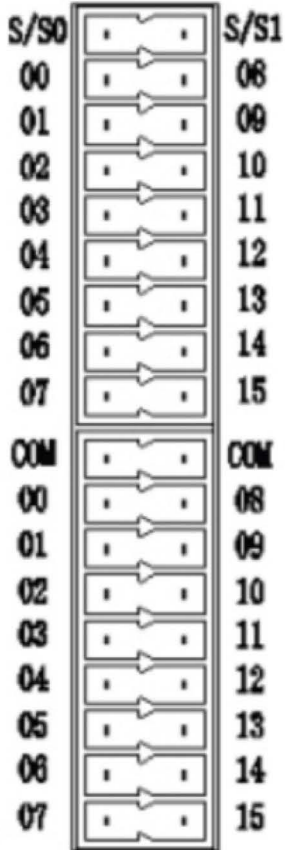
10 Разъем питания

11 Батарейный отсек

12 Шильдик

## Расположение клемм ЦПУ

MX308-CE/MX316-CE/MX332-CE



Входы 00 – 11 являются высокоскоростными, входы 12 – 15 являются обычными

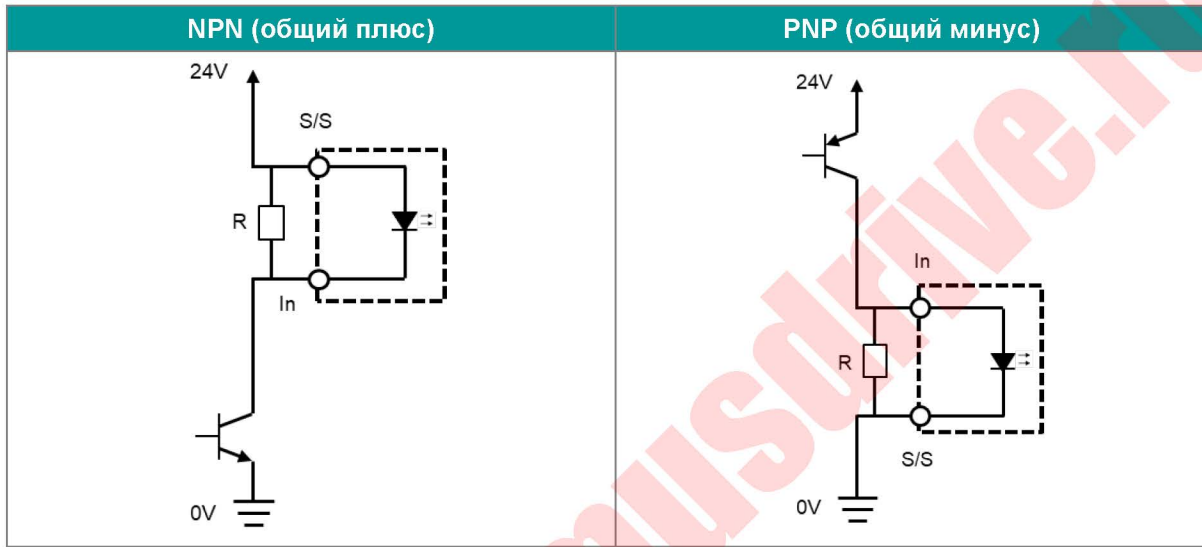
Выходы 00 – 11 являются высокоскоростными, выходы 12 -15 обычными  
(нечётный выход – импульсы, чётный – направление)

**Внимание!** При работе выхода на индуктивную нагрузку установка внешнего обратного диода на катушку является обязательной! В противном случае выход контроллера может выйти из строя уже при первом выключении индуктивной нагрузки (катушки реле/контактора).

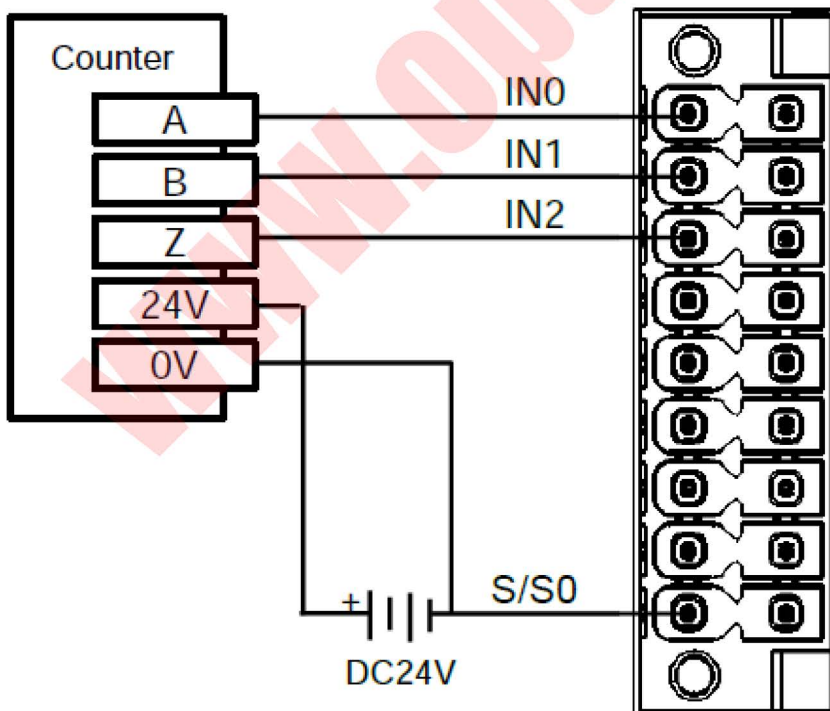
## Схемы подключения МХ308-СЕ/МХ316-СЕ/МХ332-СЕ

Общие рекомендации:

1. При подключении к контроллеру источника высокочастотного сигнала типа открытый коллектор для достижения частоты 200 кГц необходимо подключить параллельно входу и точкой S/S резистор номиналом 3 Вт/470 Ом или 2 Вт/1 кОм. На схеме ниже обозначен как R. Вход обозначен как In.
2. Если в качестве источника сигнала используется тип push-pull, то резисторы не нужны.

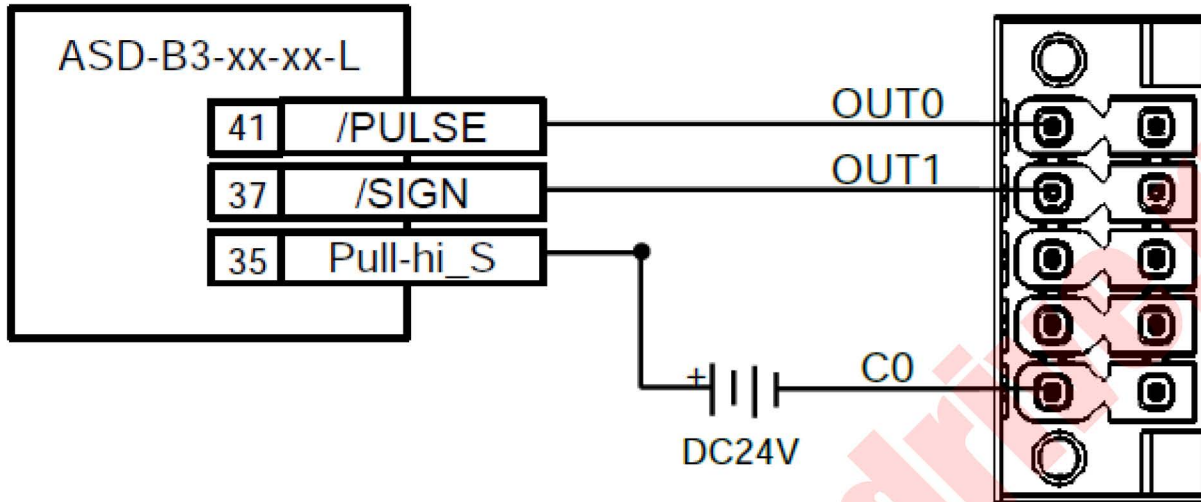


Подключение входов в режиме высокоскоростного счётчика

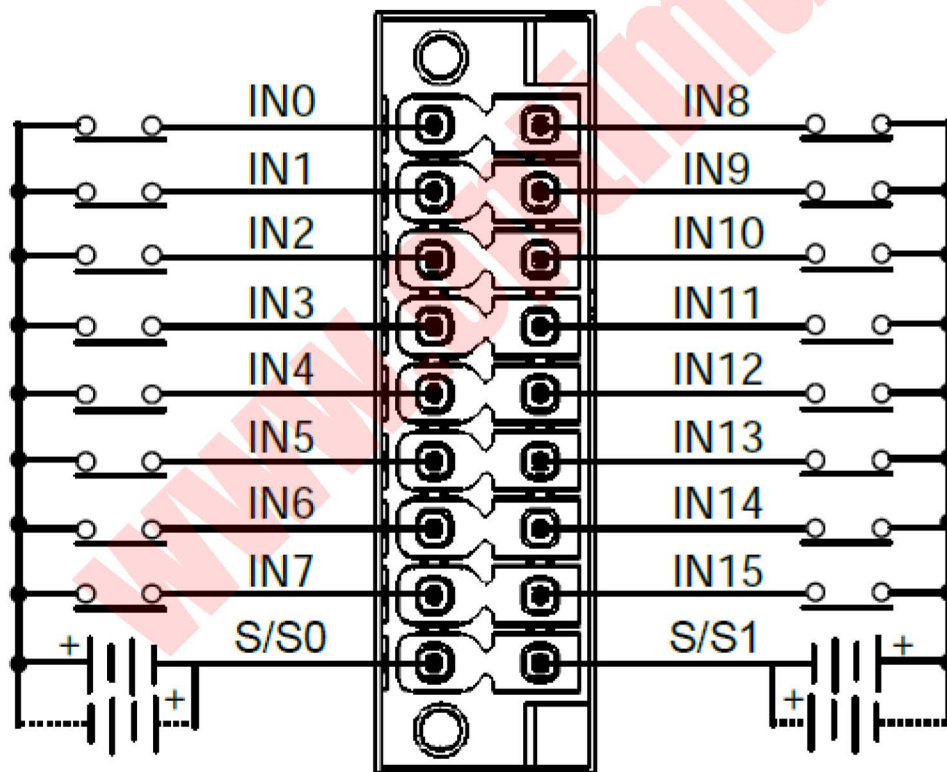


*Подключение выходов в импульсном режиме*

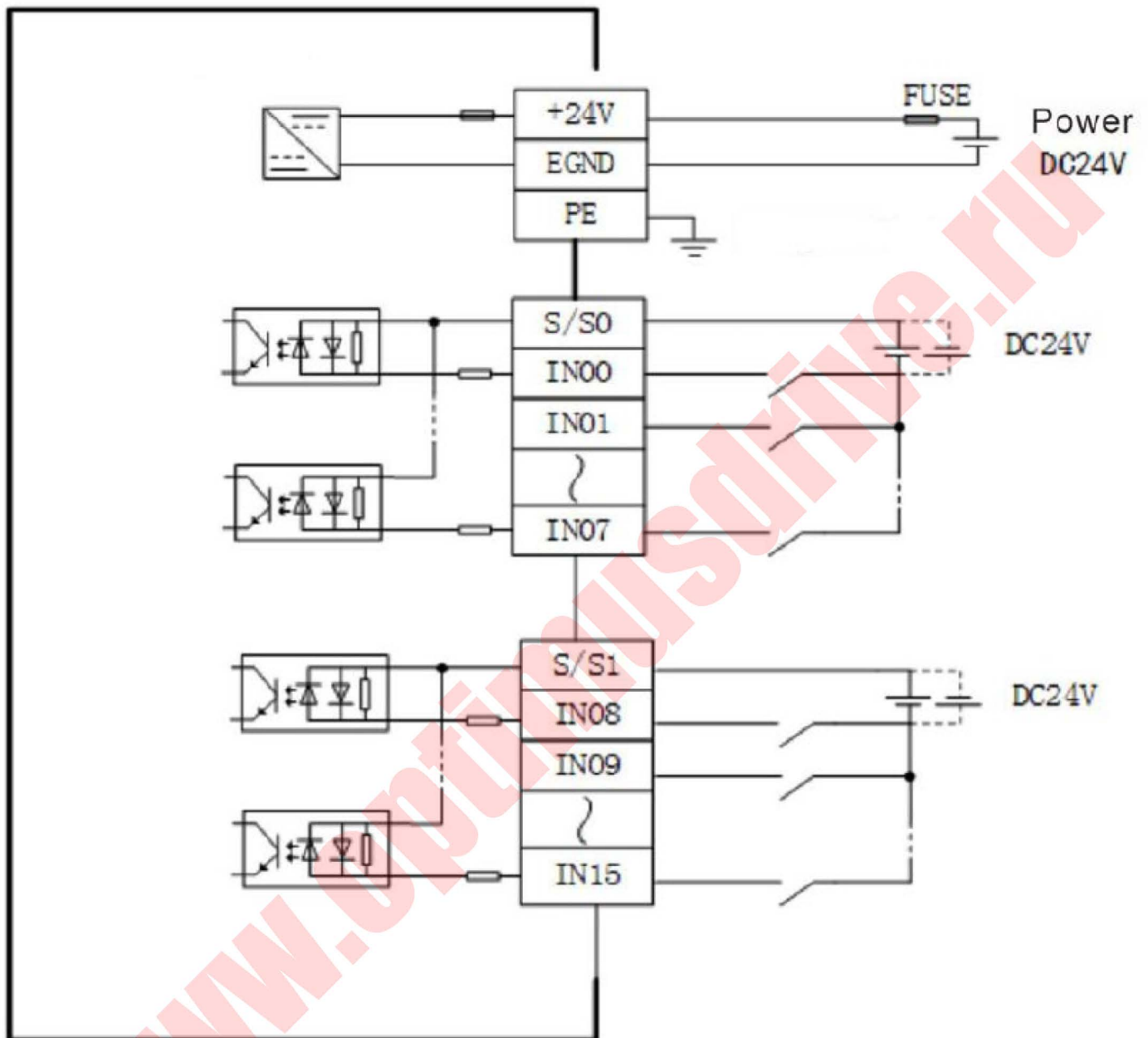
(на примере выдачи импульсов на сервопривод Delta ASD)



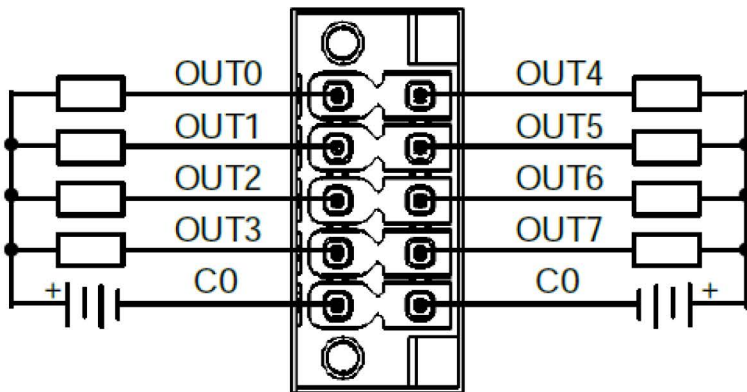
*Подключение входов в обычном режиме*



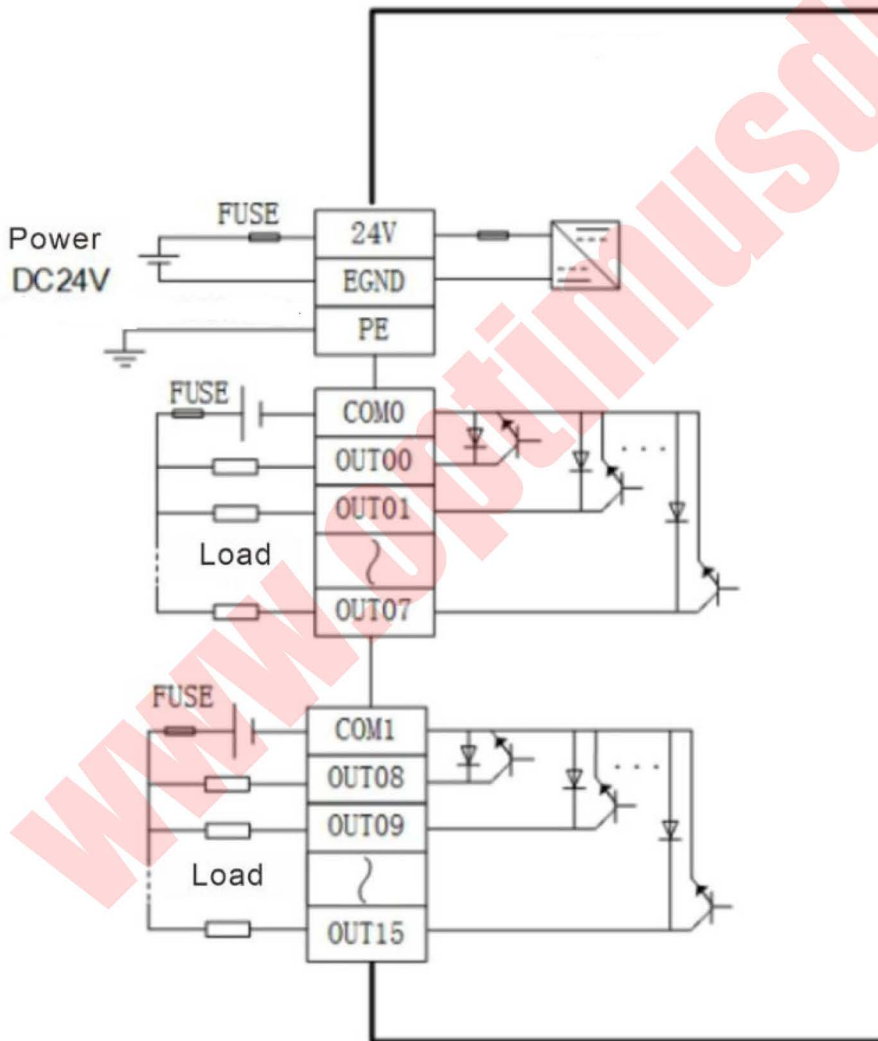
Эквивалентная схема входов:



Подключение выходов типа NPN в обычном режиме



Эквивалентная схема выходов:



**Внимание!** При работе выхода на индуктивную нагрузку установка внешнего обратного диода на катушку является обязательной! В противном случае выход контроллера может выйти из строя уже при первом выключении индуктивной нагрузки (катушки реле/контактора).

## Расположение интерфейсов MX308-CE/MX316-CE/MX332-CE

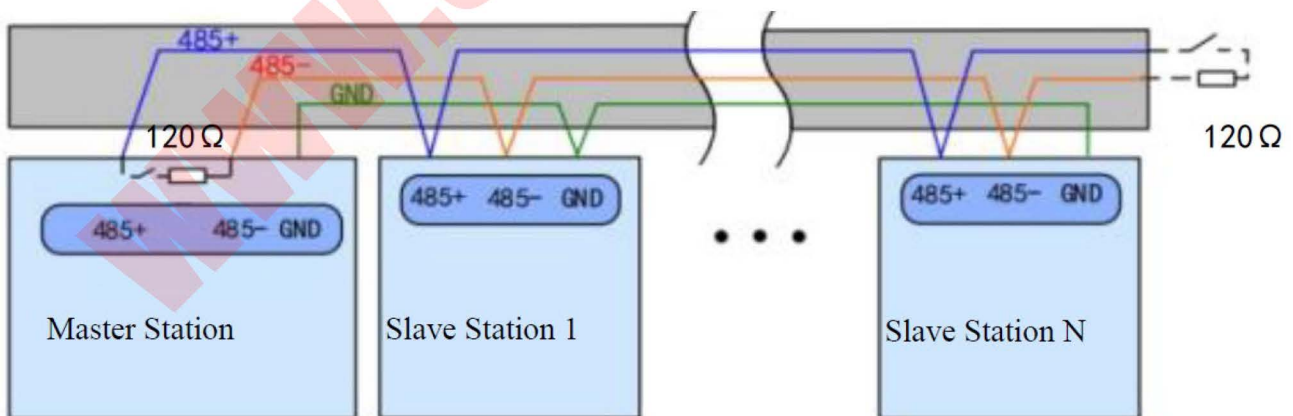
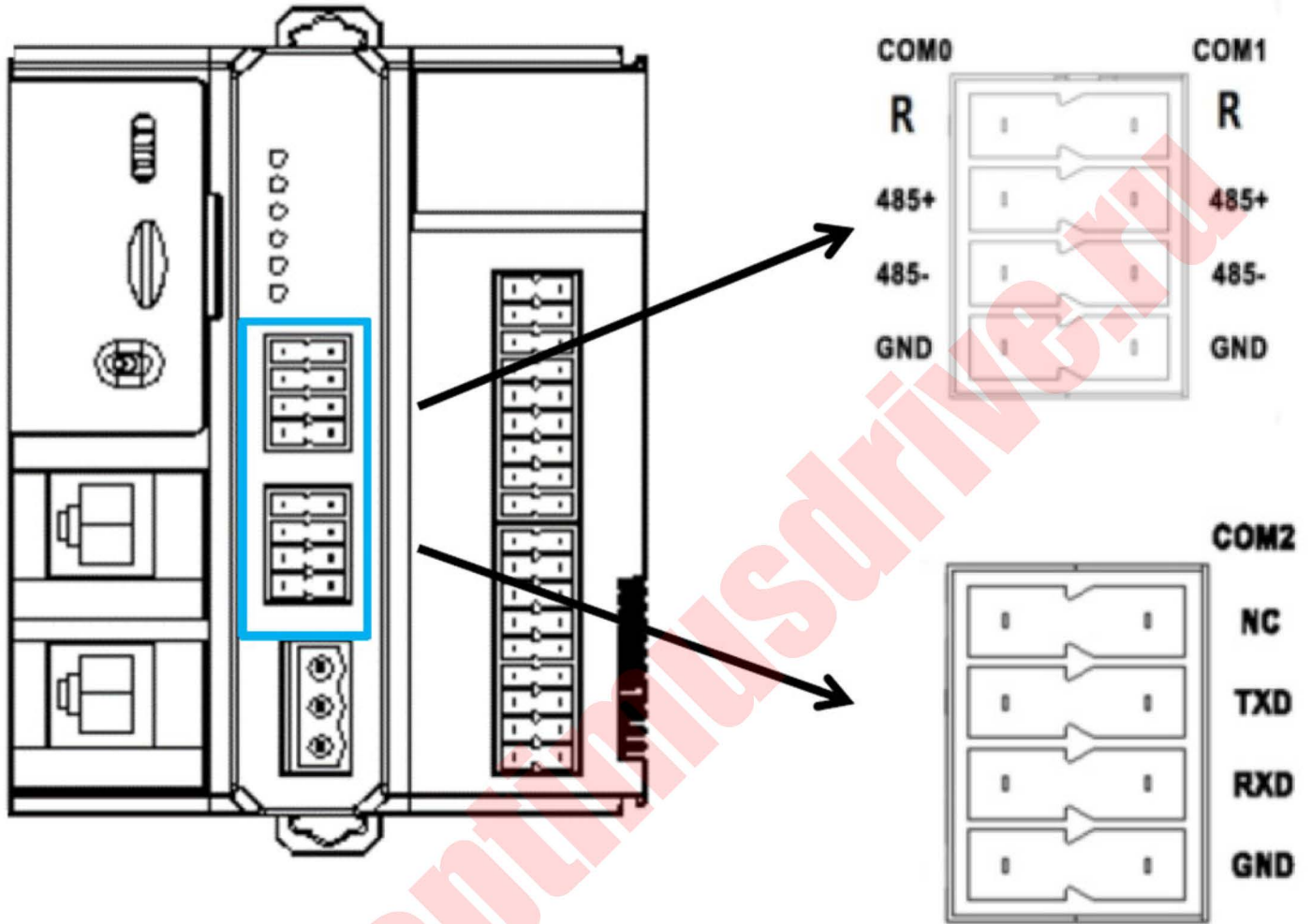
EtherCAT



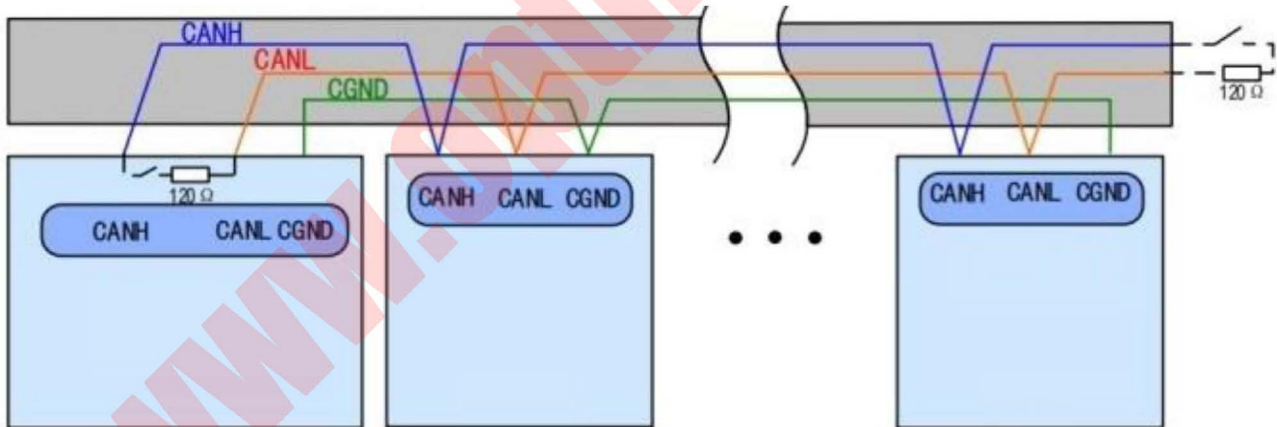
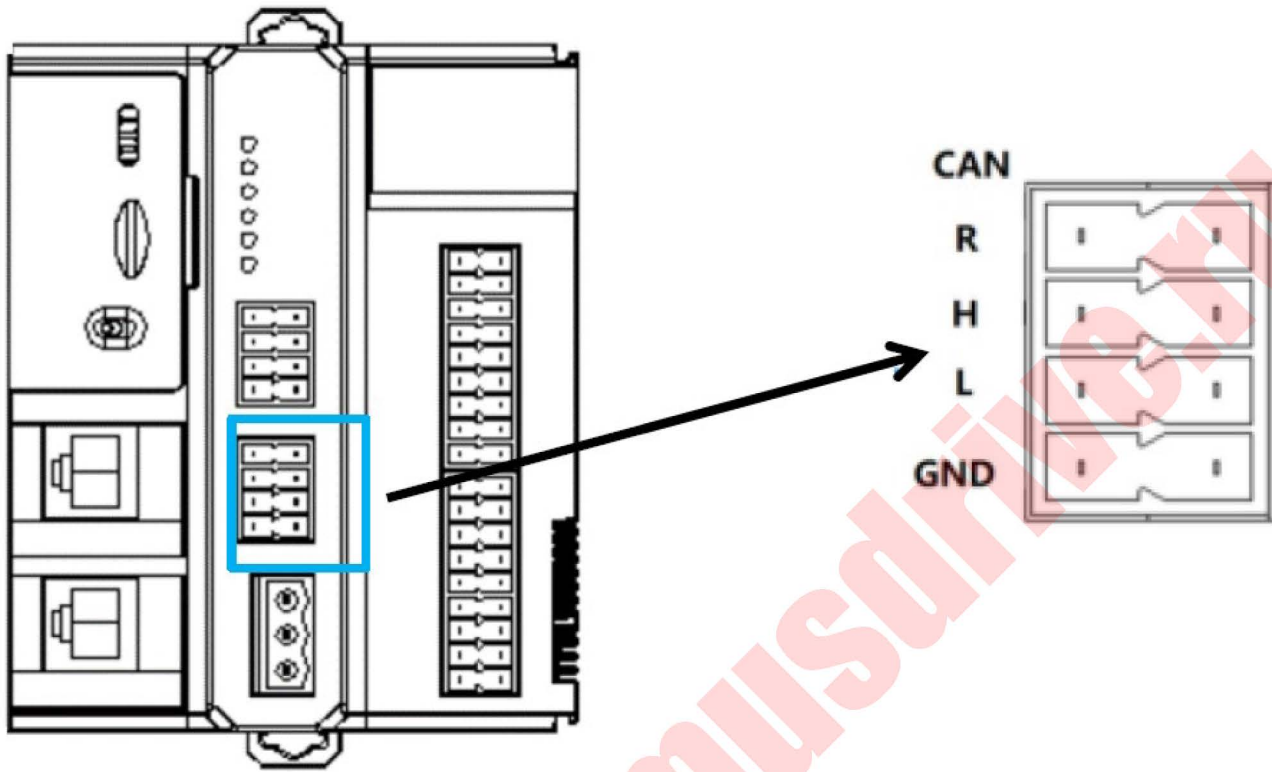
No.	Наименование	Описание
1	TX data+	Send data+
2	TX data-	Send data-
3	RX data+	Receive data+
4	/	/
5	/	/
6	RX data-	Receive data-
7	/	/
8	/	/



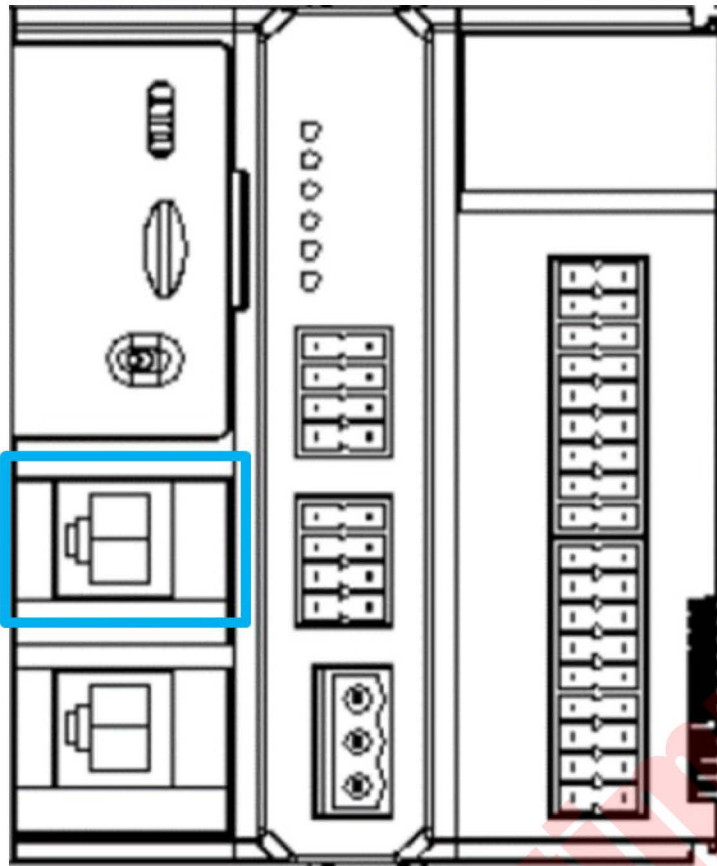
RS232/RS485



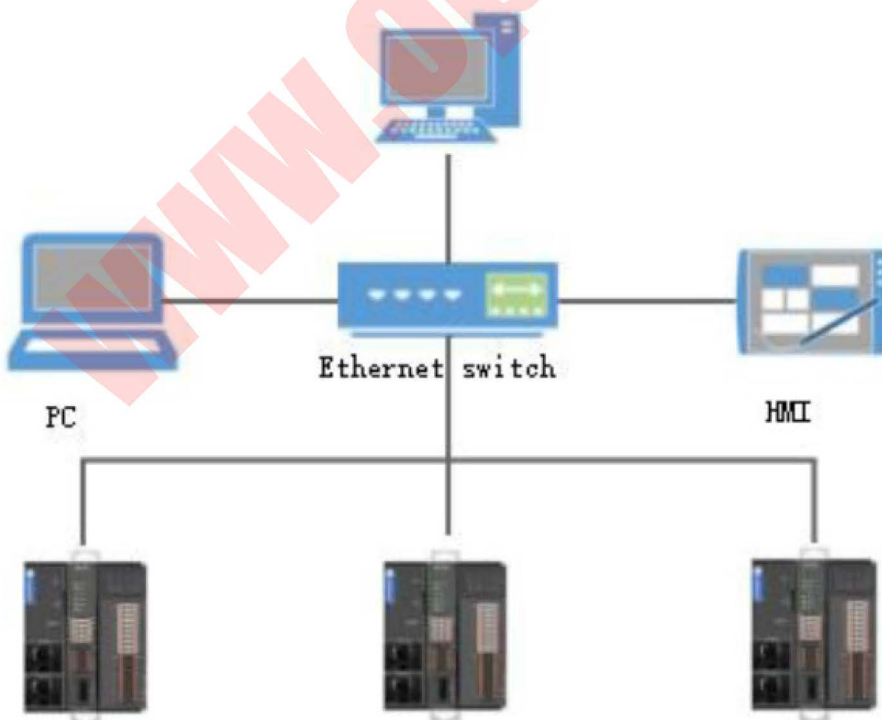
CAN Bus



## Ethernet



Контроллер может быть подключен к сети Ethernet напрямую или через коммутатор.



## Спецификация источника питания

Контроллеры требуют питание 24 VDC от стабилизированного источника питания. В комплекте с контроллером идёт клеммник для подключения внешнего источника питания, мощность которого определяется количеством и составом подключенных к контроллеру модулей расширения.

WWW.OPTIMUSDRIVE.RU

## Установка модулей расширения

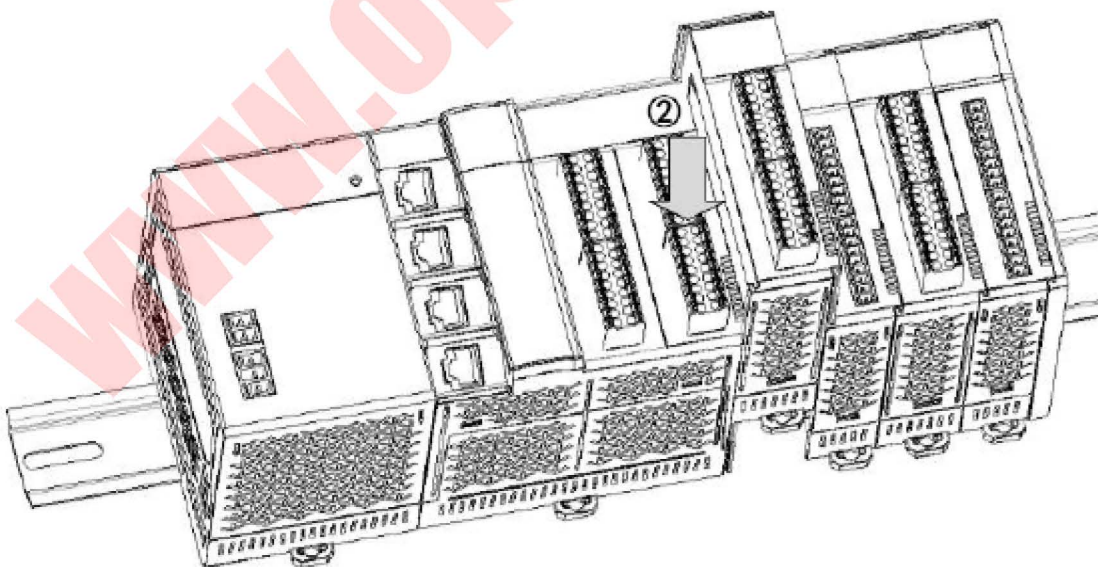
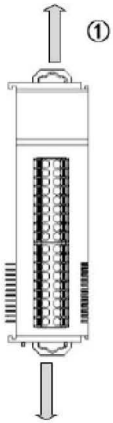
Модули расширения устанавливаются по направляющим вплотную к ЦПУ и далее модуль к модулю. Благодаря такому монтажу, для смены модуля не требуется раздвигать всю сборку. Достаточно просто откинуть разъём извлечь старый модуль вставить новый и установить обратно разъём с проводами. Для монтажа сигнальных проводов на модулях используются съёмные клеммники с пружинными зажимами.

На модулях, как и на ЦПУ, используется удобная двойная защёлка для монтажа на ДИН-рейку. Для установки модуля необходимо:

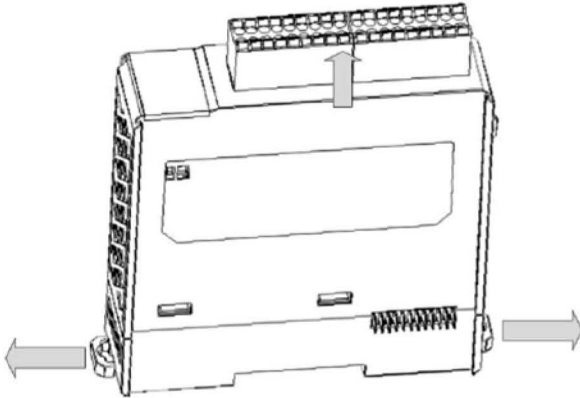
Раздвинуть фиксаторы вверх и вниз соответственно

Вставить модуль по направляющим до упора на ДИН-рейку

Защёлкнуть оба фиксатора (сверху и снизу)



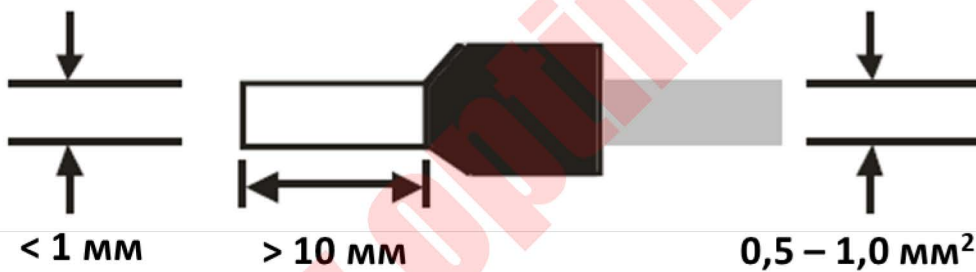
Для извлечения модуля необходимо раздвинуть фиксаторы и потянуть модуль на себя:



### Рекомендации по использованию проводов и наконечников

На контроллерах МХ300 и модулях расширения установлены пружинные клеммники, которые позволяют производить монтаж проводов без наконечников. Рекомендуются строго гибкие (многопроволочные) провода сечением 0,5 – 1,0 мм<sup>2</sup>.

В случае использования наконечников рекомендуются наконечники без юбки. Если Вы принципиально используете наконечники с юбкой, то длина рабочей части наконечника должна быть не менее 10 мм. Диаметр после обжима не более 1 мм.



## Спецификация модулей дискретных входов-выходов

### Спецификация дискретных входов на модулях расширения:

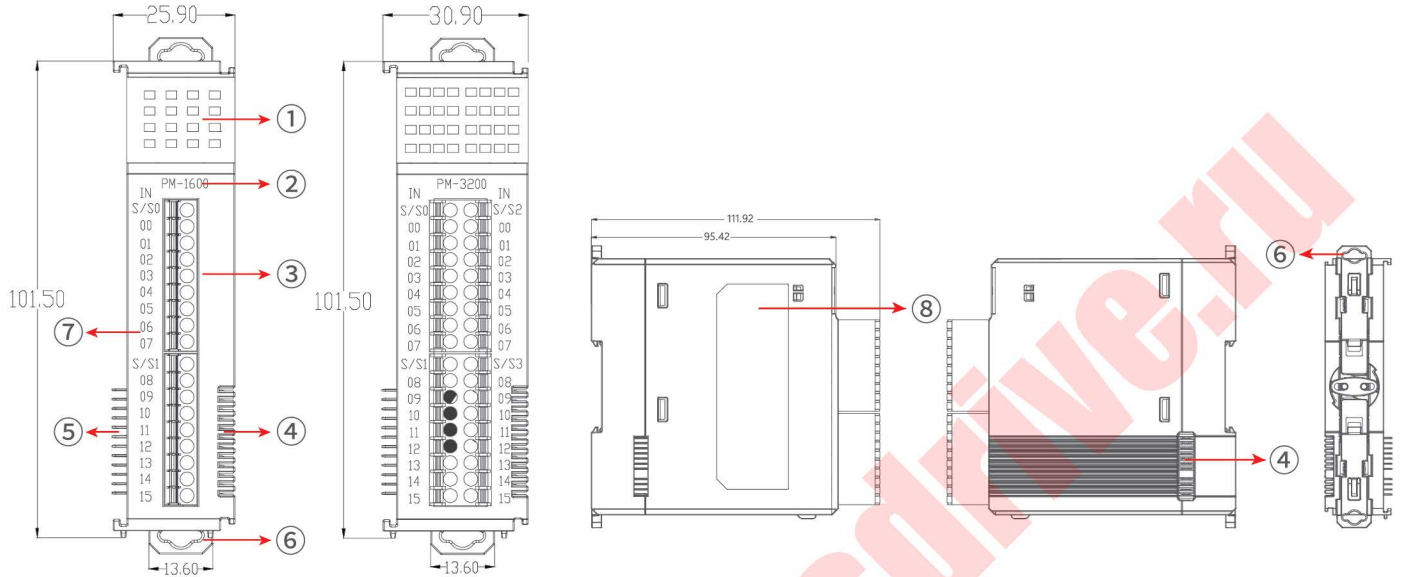
Параметр	Модель	MX16DI
Количество входов		16
Тип соединения		Съемный пружинный клеммник
Тип входа		Дискретный вход
Форма входа		Постоянный ток (NPN/PNP, две общие точки SS0/SS1) SS0 для IN0~IN7, SS1 для IN8~IN15
Напряжение/ ток		24 VDC, 5 mA
Уровень вкл/выкл	OFF→ON	>15 VDC
	ON→OFF	<5 VDC
ON/OFF		20мкс/50мкс
Входное сопротивление		4.7 kΩ
Тип входного сигнала		Потенциальный сигнал Sinking: SS0/SS1 подключены к 24V+ Sourcing: SS0/SS1 подключены к 0V
Электрическая изоляция		оптопары
Индикация		Когда оптопара активна, светодиод входа включен
Потребляемая мощность		0,7 Вт
Габаритные размеры		ВхШхГ: 100.00 x 25.90 x 98.5 мм

### Спецификация дискретных выходов на модулях расширения:

Параметр	Модель	MX16DOT	MX16DOP	MX16DOR
Количество выходов		16		
Тип соединения		Съемный пружинный клеммник		
Тип выхода		NPN (Sinking)	PNP (Sourcing)	Реле
Напряжение		5~30 VDC		250VAC/30VDC
Макси- мальная нагрузка	Активная	500 мА(канал)		2А(канал)
	Индуктивная	-		
	Лампочка	-		
Максимальная выходная частота		1 КГц	1 КГц	1 Гц
Время переключения	ON OFF	20мкс/50мкс	15мкс/40мкс	15мс/15мс
Потребляемая мощность		1.66 Вт	3.3 Вт	3.0 Вт
Габаритные размеры		ВхШхГ: 100.00 x 25.90 x 98.5 мм		

## Внешний вид и размеры дискретных модулей расширения

### MX16DI/MX16DOT/MX16DOP/MX16DOR

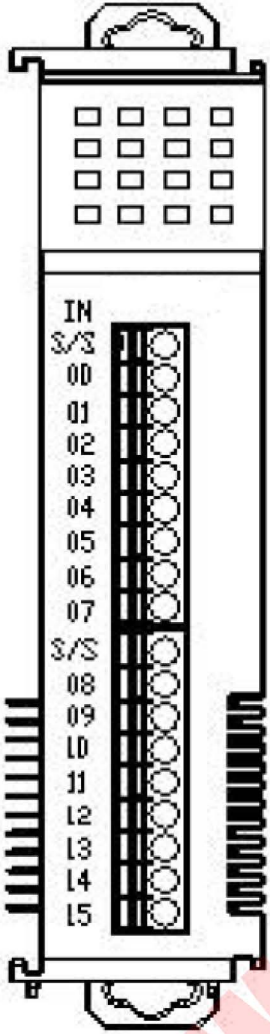


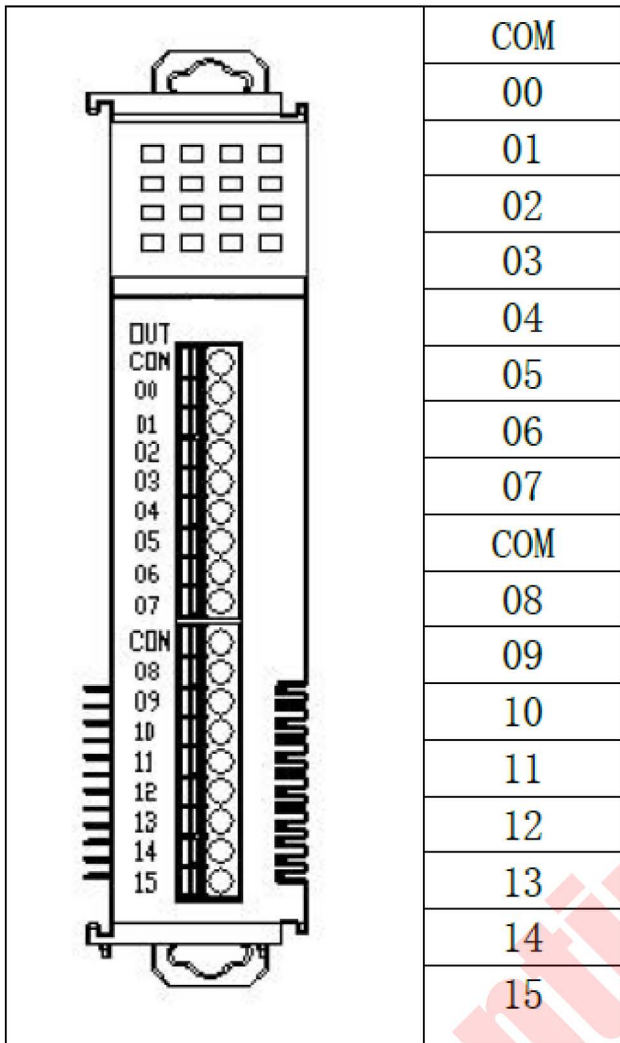
- 1 Светодиоды для входов/выходов: Горят при наличии сигнала на входе или выходе
- 2 Название модели
- 3 Съёмный клеммник с пружинными клеммами
- 4 Разъем подключения модуля расширения
- 5 Разъем подключения модуля расширения
- 6 Крепление на DIN-рейку
- 7 Названия клемм
- 8 Шильдик



## Расположение клемм дискретных модулей расширения

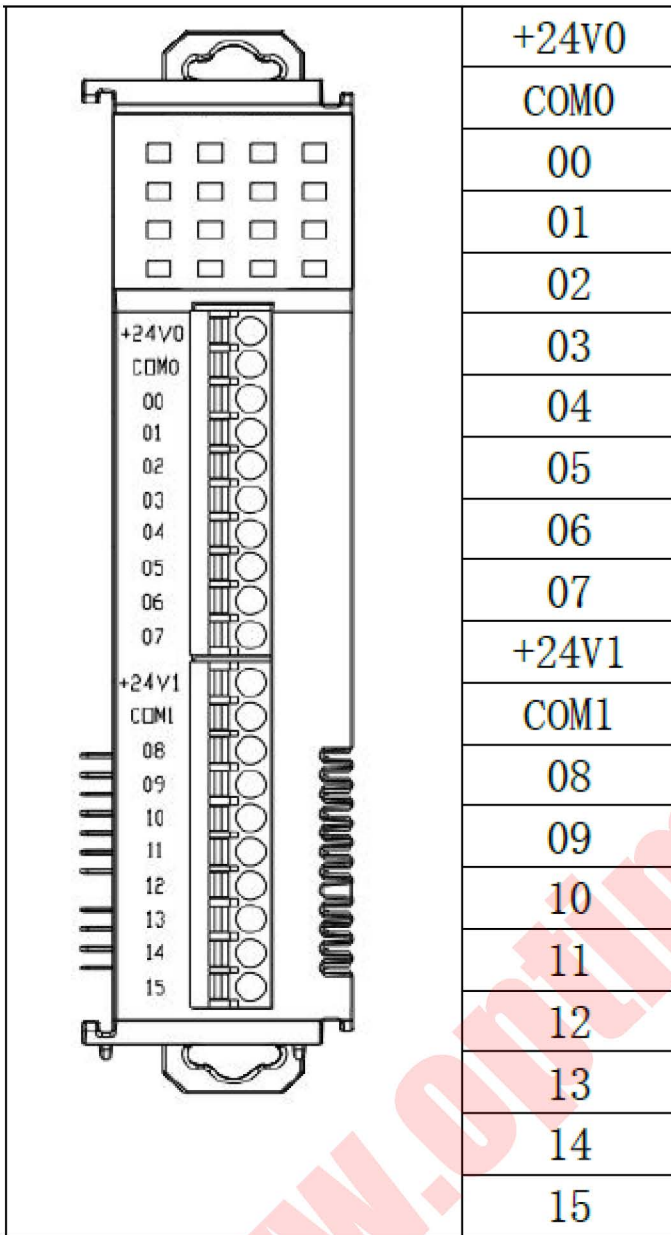
MX16DI

	S/S 0
	00
	01
	02
	03
	04
	05
	06
	07
	S/S 1
	08
	09
	10
	11
	12
	13
14	
15	

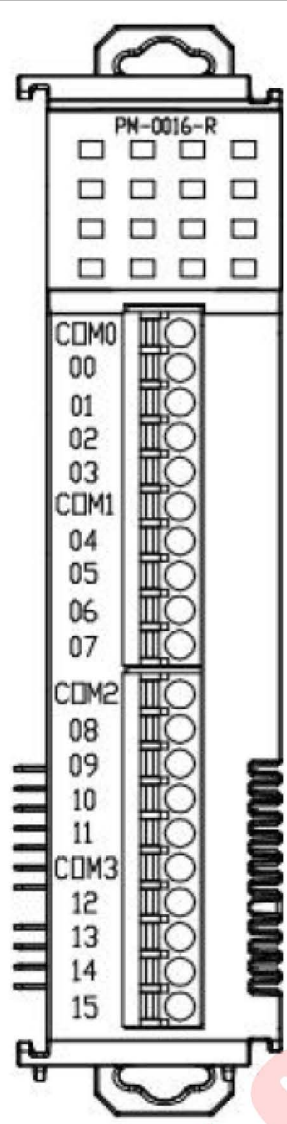
**MX16DOT**


www.optimusdrive.ru

MX16DOP

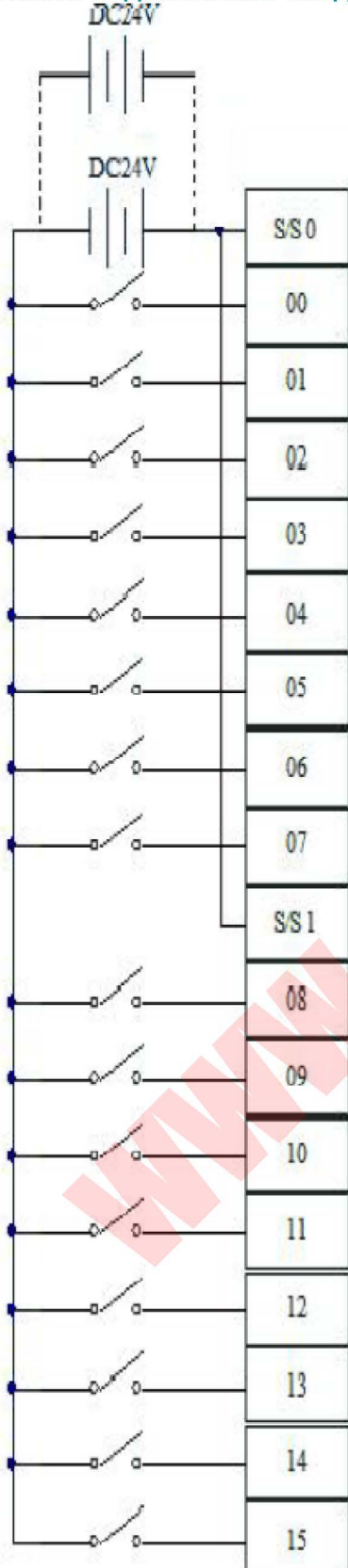


**MX16DOR**

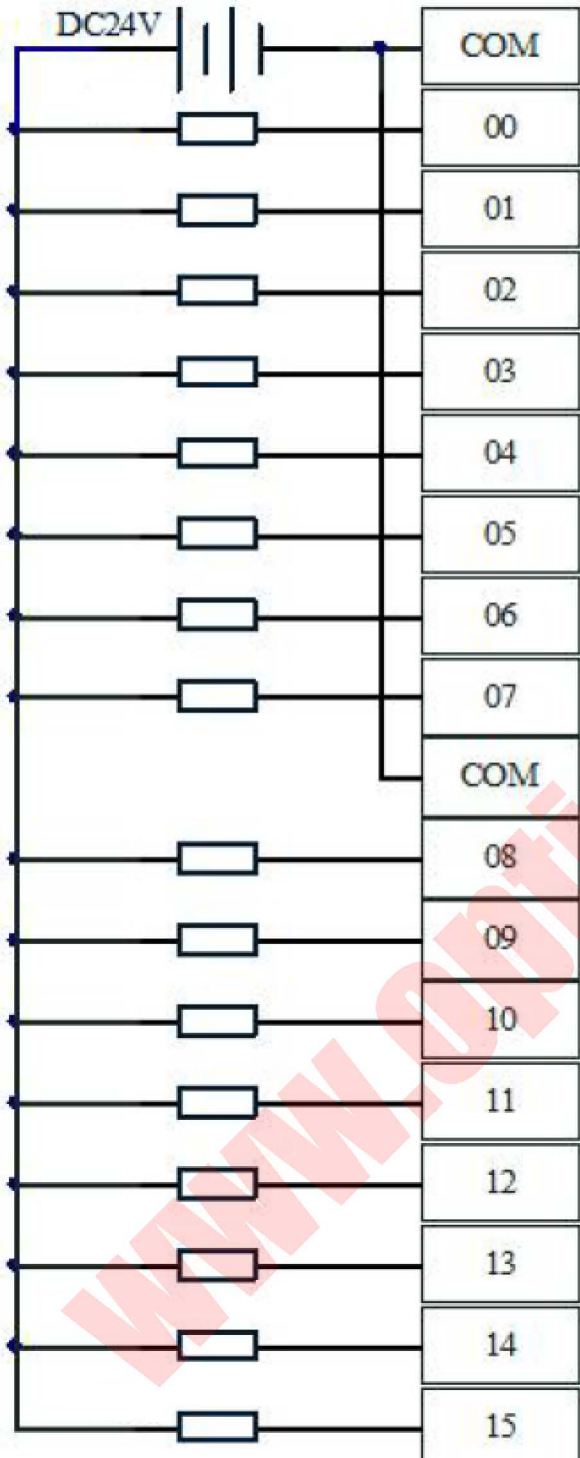
	COM0
	00
	01
	02
	03
	COM1
	04
	05
	06
	07
	COM2
	08
	09
	10
	11
	COM3
12	
13	
14	
15	

## Схемы подключения дискретных входов-выходов модулей расширения

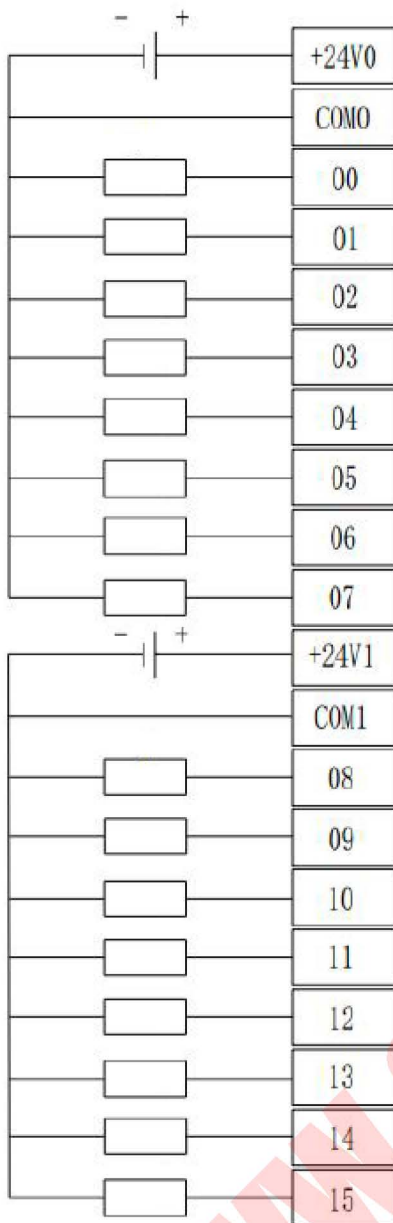
Схемы подключения входов на модуле расширения MX16DI



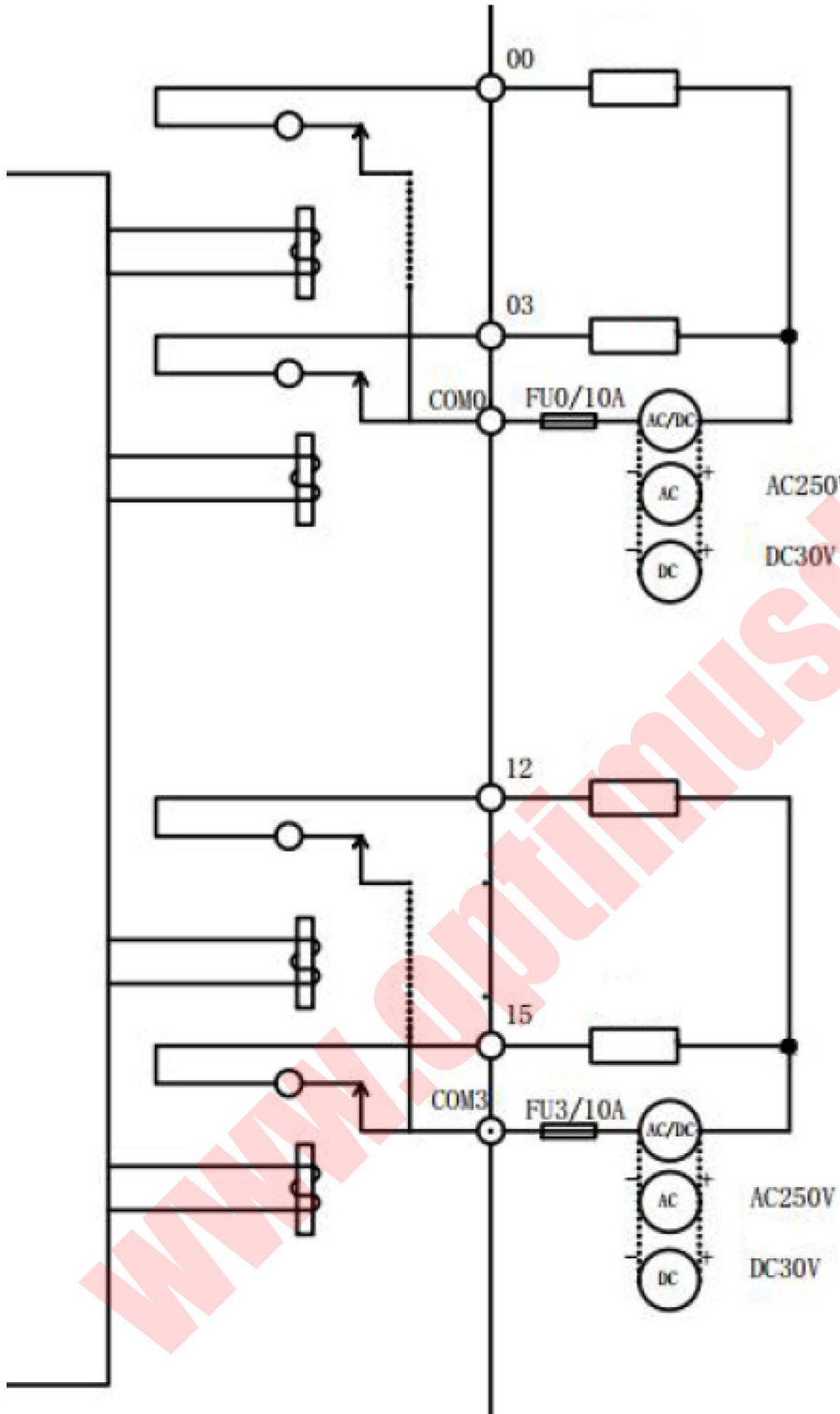
## Схемы подключения выходов на модуле расширения MX16DOT



## Схемы подключения выходов на модуле расширения MX16DOP



Схемы подключения выходов на модуле расширения MX16DOR





## Спецификация модулей аналоговых входов-выходов

### MX04AD

#### Общие характеристики

Модуль	MX04AD
Число входов	4
АЦП	Вход по напряжению / вход по току
Напряжение питания	24 VDC
Разъем	Разъемный клеммный блок
Время преобразования	1 мс/4 канала
Габаритные размеры	ВхШхГ: 100.0 x 23.0 x 98.5 мм
Потребляемая мощность	2.0 Вт
Рабочая температура	0-55°C

#### Характеристики преобразования входных сигналов

АЦП	Вход по напряжению				
Рабочий диапазон	-10 В~10 В	0 В~10 В	±5 В	0 В~5 В	1 В~5 В
Аппаратный диапазон	-10.2 В ~10.2 В	-0.2 В ~10.2 В	-5.1 В ~5.1 В	-0.1 В ~5.1 В	0.2 В ~5.1 В
Диапазон цифровой шкалы	-32000 ~ 32000				
Погрешность осреднения при температуре 25°C	±0.2%				
Погрешность осреднения при полном температурном диапазоне 0-55°C	±0.3%				
Аппаратное разрешение	16 бит				
Входной импеданс	>1 MΩ				

АЦП	Вход по току		
Рабочий диапазон	±20 мА	0 мА~20 мА	4 мА~20 мА
Аппаратный диапазон	-20.2 мА~20.2 мА	-0.2 мА~20.2 мА	3.8 мА~20.2 мА
Диапазон цифровой шкалы	-32000 ~ 32000		
Погрешность осреднения при температуре 25°C	±0.3%		
Погрешность осреднения при полном температурном диапазоне 0-55°C	±0.4%		
Аппаратное разрешение	16 бит		
Входной импеданс	250 Ω		

**MX04DA**
**Общие характеристики**

Модуль	<b>MX04DA</b>
Число выходов	4
ЦАП	Выход по напряжению / выход по току
Напряжение питания	24 VDC
Разъем	Разъемный клеммный блок
Время преобразования	1 мс/4 канала
Габаритные размеры	ВхШхГ: 100.0 x 23.0 x 98.5 мм
Потребляемая мощность	2.0 Вт
Рабочая температура	0-55°C

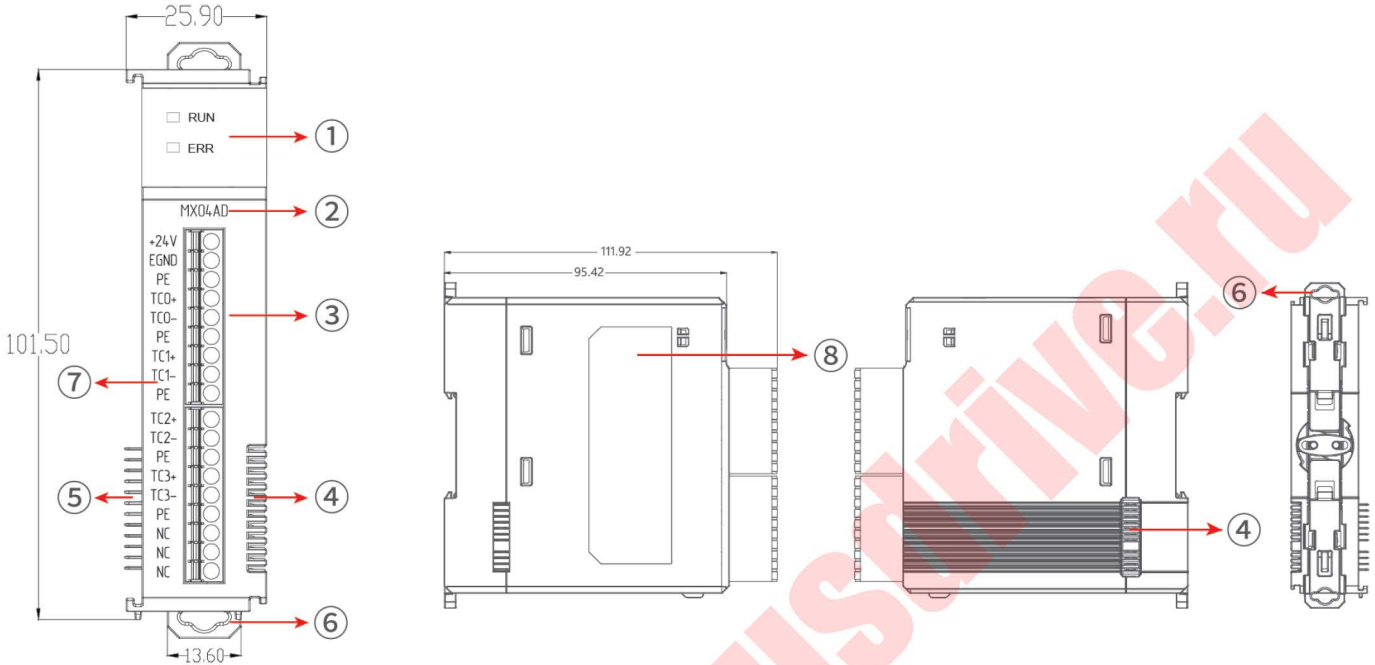
**Характеристики преобразования входных сигналов**

ЦАП	Выход по напряжению				
Рабочий диапазон	-10 В~10 В	0 В~10 В	±5 В	0 В~5 В	1 В~5 В
Аппаратный диапазон	-10.2 В ~10.2 В	-0.2 В ~10.2 В	-5.1 В ~5.1 В	-0.1 В ~5.1 В	0.2 В ~5.1 В
Диапазон цифровой шкалы	-32000 ~ 32000				
Погрешность осреднения при температуре 25°C	±0.2%				
Погрешность осреднения при полном температурном диапазоне 0-55°C	±0.3%				
Аппаратное разрешение	16 бит				
Входной импеданс нагрузки	1 КΩ ~ 1 МΩ				

ЦАП	Выход по току	
Рабочий диапазон	0 мА~20 мА	4 мА~20 мА
Аппаратный диапазон	-0.2 мА~20.2 мА	3.8 мА~20.2 мА
Диапазон цифровой шкалы	-32000 ~ 32000	
Погрешность осреднения при температуре 25°C	±0.3%	
Погрешность осреднения при полном температурном диапазоне 0-55°C	±0.4%	
Аппаратное разрешение	16 бит	
Входной импеданс нагрузки	100 - 500 Ω	

## Внешний вид и размеры аналоговых модулей

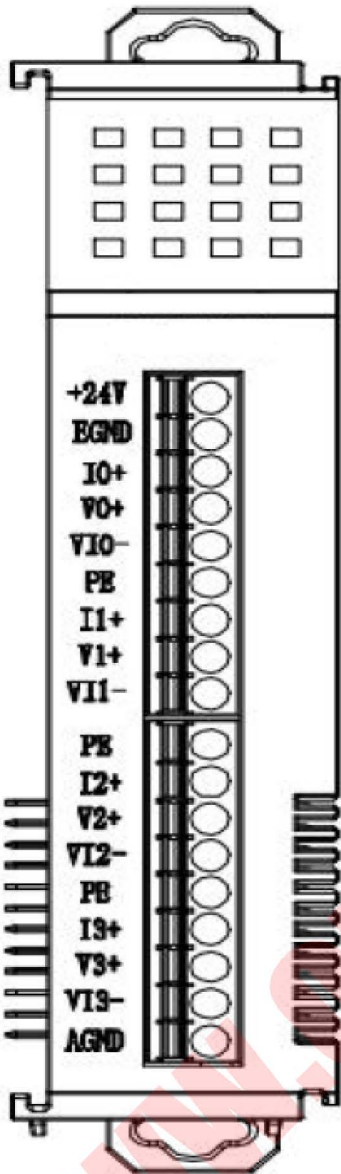
### MX04AD/MX04DA



- 1 Светодиоды: RUN: работа (зеленый), ERR: ошибка (красный)
- 2 Название модели
- 3 Съёмный клеммник с пружинными клеммами
- 4 Разъем подключения модуля расширения
- 5 Разъем подключения модуля расширения
- 6 Крепление на DIN-рейку
- 7 Названия клемм
- 8 Шильдик

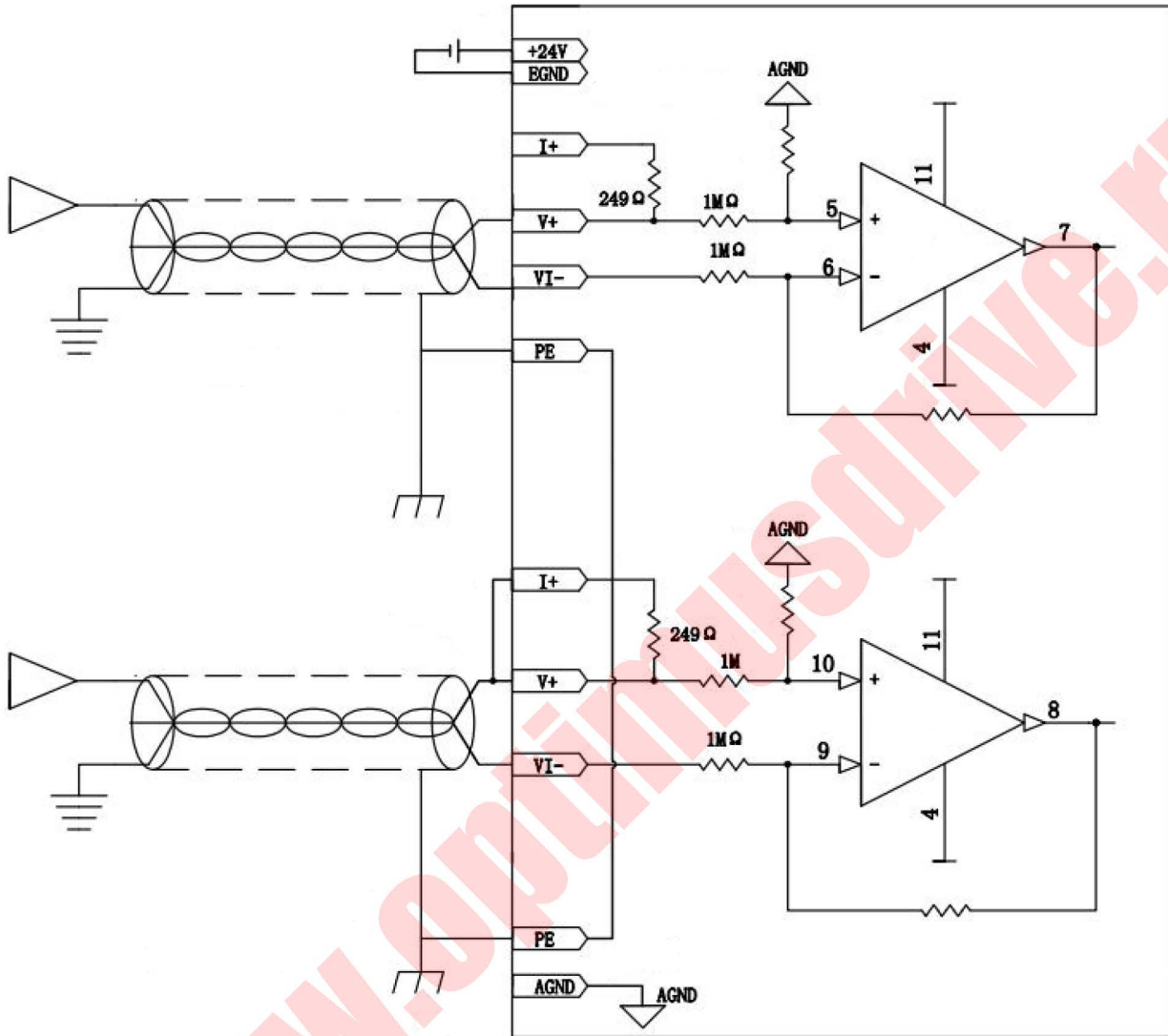
## Расположение клемм аналоговых модулей

MX04AD/MX04DA

 <p> <b>+24V</b>  <b>EGND</b>  <b>I0+</b>  <b>V0+</b>  <b>VI0-</b>  <b>PE</b>  <b>I1+</b>  <b>V1+</b>  <b>VI1-</b>  <b>PE</b>  <b>I2+</b>  <b>V2+</b>  <b>VI2-</b>  <b>PE</b>  <b>I3+</b>  <b>V3+</b>  <b>VI3-</b>  <b>AGND</b> </p>	+24V
	EGND
	I0+
	V0+
	VI0-
	PE
	I1+
	V1+
	VI1-
	PE
	I2+
	V2+
	VI2-
	PE
	I3+
	V3+
VI3-	
AGND	

## Схемы подключения аналоговых входов-выходов

### Схемы подключения входов на модуле расширения MX04AD



Модуль требует внешнего питания. Подключите 24 VDC к клеммам +24V и EGND.

Для подключения датчика используйте экранированную витую пару, экран которой нужно подключить к сигнальному заземлению.

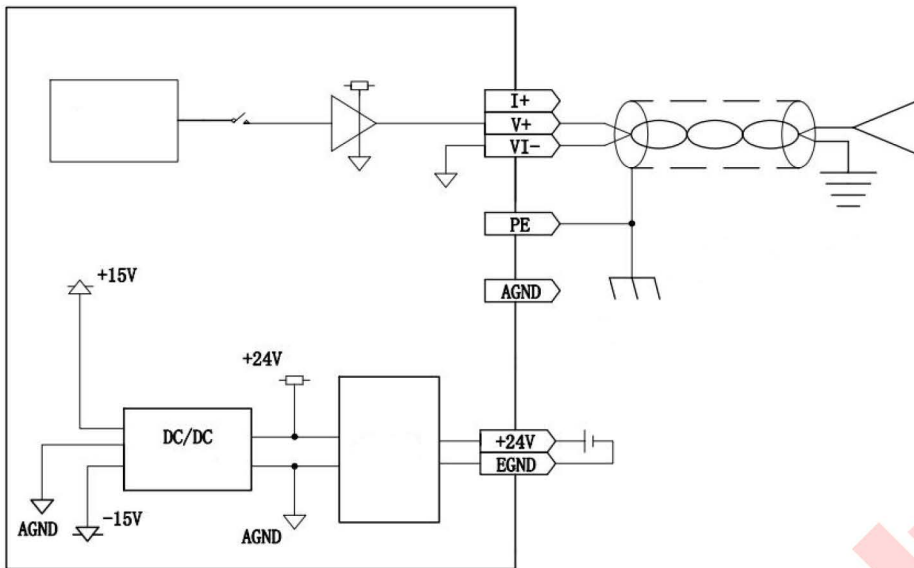
Модуль необходимо установить на хорошо заземленную металлическую ДИН-рейку. В верхней части защёлки имеется металлический вывод, который при защёлкивании модуля на ДИН-рейке в ходит в контакт с ней. Обе клеммы PE соединены с этим выводом и между собой. Экраны можно подсоединить к клеммам PE при наличии хорошего контакта модуля и ДИН-рейки. При необходимости провод заземления можно подсоединить на одну из клемм PE.

При подключении токового сигнала поставьте перемычку между клеммами V+ и I+.

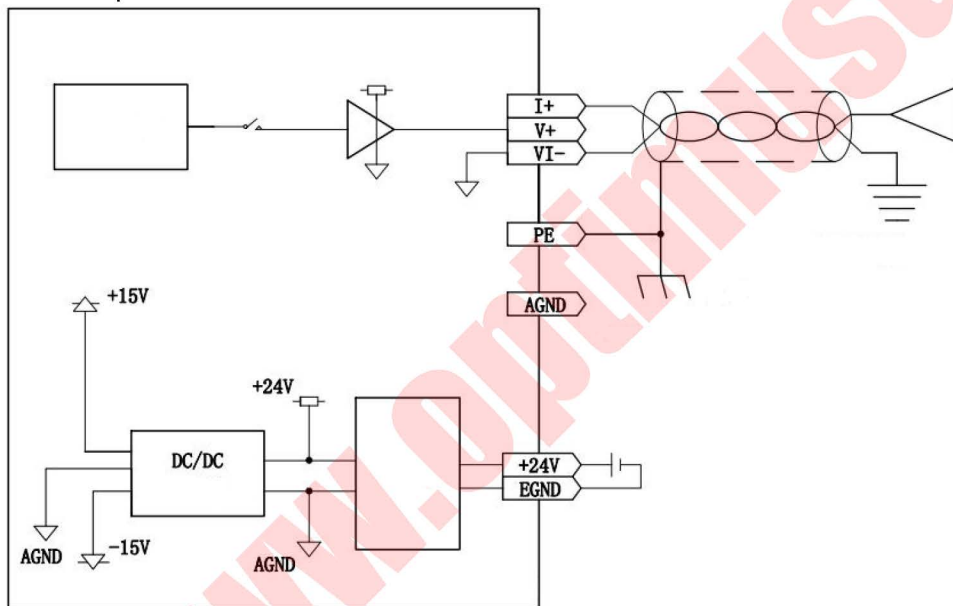
Когда входной сигнал является дифференциальным сигналом, клемму «AGND» можно подключить к аналоговой земле совместимого устройства (датчика) для устранения синфазных различий между устройствами (по сути это балластный провод). Данная мера повысит точность измерений.

Схемы подключения выходов на модуле расширения MX04DA

Потенциальный режим



Токовый режим



Модуль требует внешнего питания. Подключите 24 VDC к клеммам +24V и EGND.

Для подключения датчика используйте экранированную витую пару, экран которой нужно подключить к сигнальному заземлению.

Модуль необходимо установить на хорошо заземленную металлическую ДИН-рейку. В верхней части защёлки имеется металлический вывод, который при защёлкивании модуля на ДИН-рейке в ходит в контакт с ней. Обе клеммы PE соединены с этим выводом и между собой. Экраны можно подсоединить к клеммам PE при наличии хорошего контакта модуля и ДИН-рейки. При необходимости провод заземления можно подсоединить на одну из клемм PE.

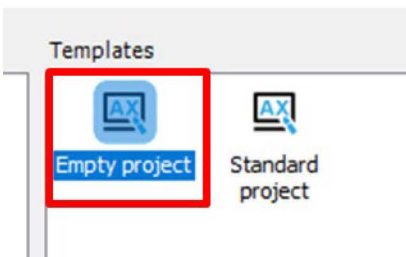
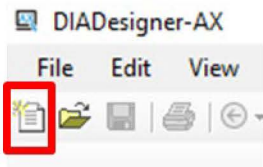
Когда входной сигнал является дифференциальным сигналом, клемму «AGND» можно подключить к аналоговой земле совместимого устройства (датчика) для устранения синфазных различий между устройствами (по сути это балластный провод). Данная мера повысит точность измерений.

## Запуск среды программирования и создание проекта

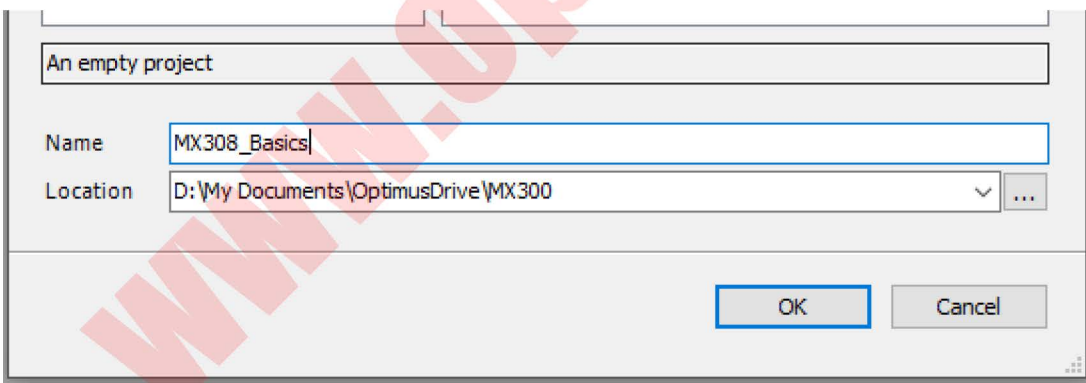
Установите на ПК (ноутбук) среду программирования DesignerAX следуя командам Мастера установки. После установки на рабочем столе появится иконка:



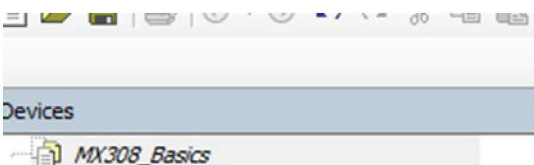
Для запуска среды программирования щёлкните дважды левой кнопкой мышки по иконке. В открывшемся рабочем окне выберите меню создания проекта и выберите создание пустого проекта «Empty Project».



Дайте название проекту:



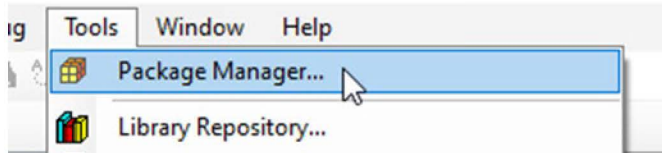
В открывшемся рабочем окне в верхнем левом углу древа проекта будет единственный пункт с названием проекта.



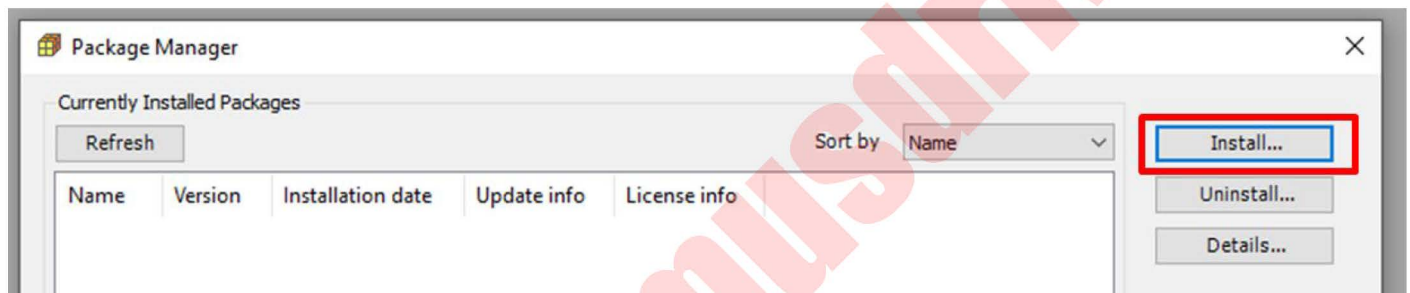
## Установка описания устройства для контроллера MX300

Для начала работы с контроллером семейства MX300 необходимо установить в среду программирования файлы с описанием устройства. В данном случае это будет пакет «**CODESYS Package**» с расширением **.package**, который содержит в себе описания контроллера и модулей расширения, библиотеки, USB драйвер и примеры.

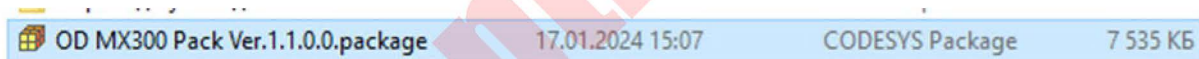
В созданном проекте войдите в меню установки пакетов устройств:



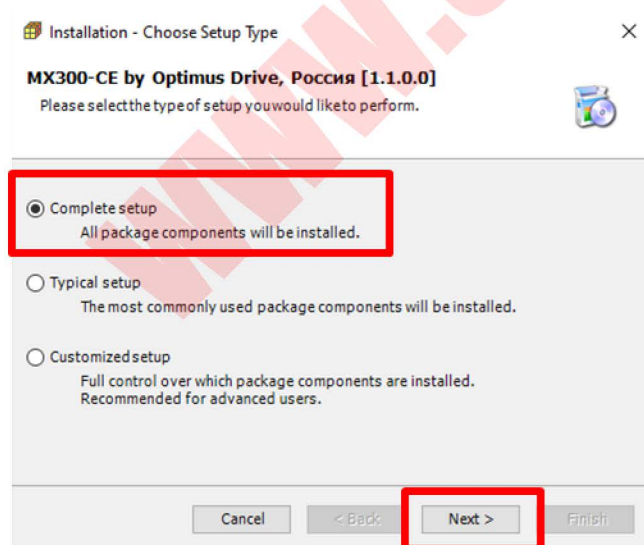
В открывшемся Мастере установки пакетов (Package Manager) нажмите кнопку **Install**:



Выберите в проводнике нужный файл:

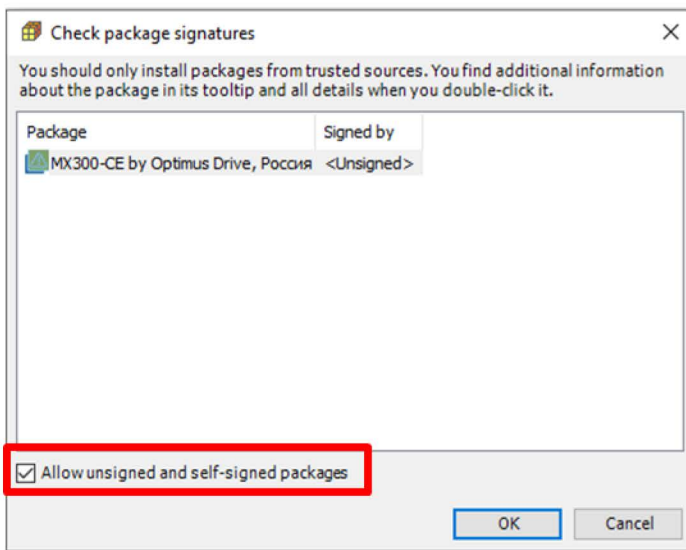


Выберите полную установку:

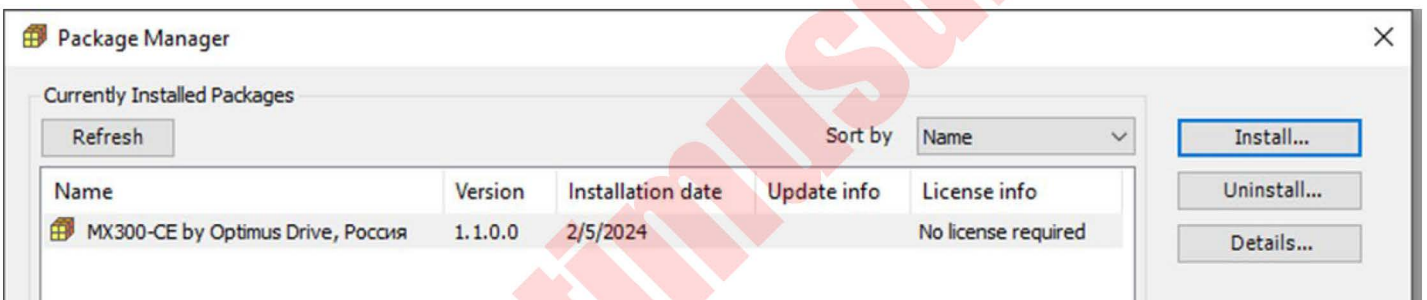




Поставьте флажок с разрешением работы без подписи файла:



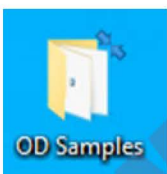
После установки появится запись в списке установленных пакетов:



Закройте менеджер пакетов.

На этом установка пакета закончена и можно переходить к добавлению устройств в проект.

На рабочем столе появится папка с примерами:

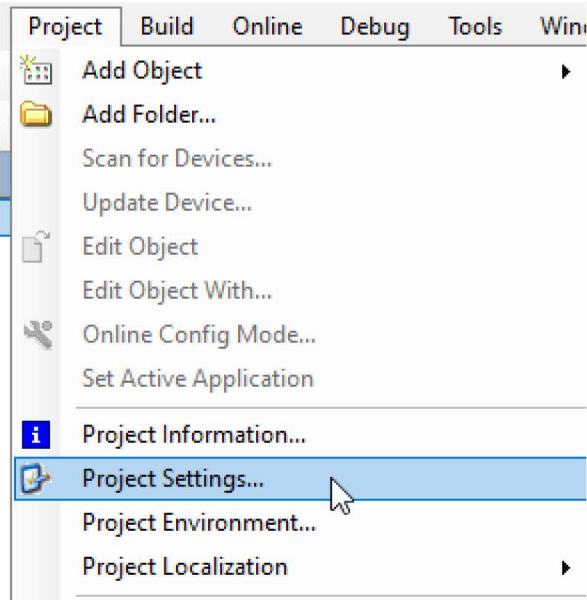


Имя	Дата изменения	Тип	Размер
Nets	05.02.2024 12:09	Папка с файлами	
MX308 HSIO.project	17.01.2024 15:03	DIADesigner-AX project	380 КБ
MX308 n R2EC Template.project	10.01.2024 15:40	DIADesigner-AX project	453 КБ
MX308 n SD700 SM QSG.project	17.01.2024 11:02	DIADesigner-AX project	436 КБ
MX308 w.Local MX_Modules.project	17.01.2024 15:04	DIADesigner-AX project	302 КБ

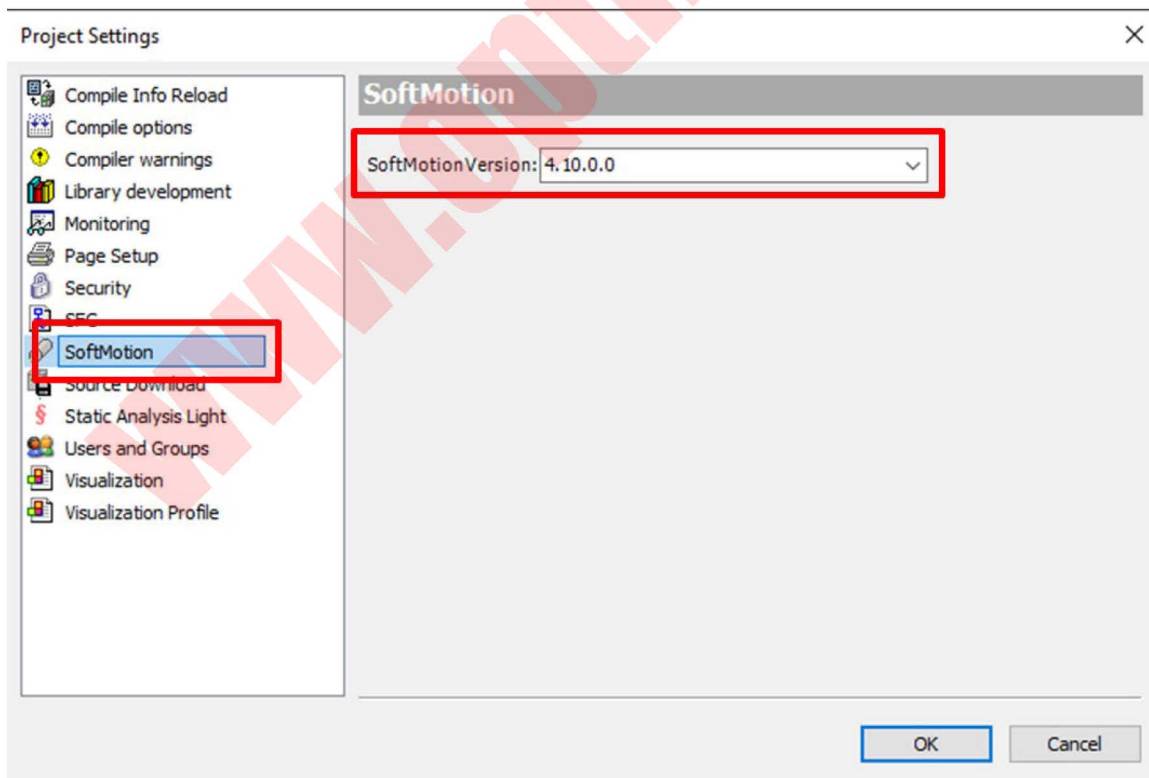
## Определение версии библиотеки 3S SoftMotion (SM3)

Для корректной работы функций управления движением необходимо выбрать правильную версию библиотеки 3S SoftMotion. Для контроллеров MX300 используется версия 4.10.0.0, для выбора которой выполните следующие действия:

Зайдите в меню **Project – Project Settings**:



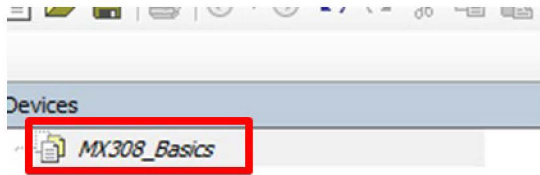
В открывшемся окне выберите пункт **SoftMotion** и версию 4.10.00:



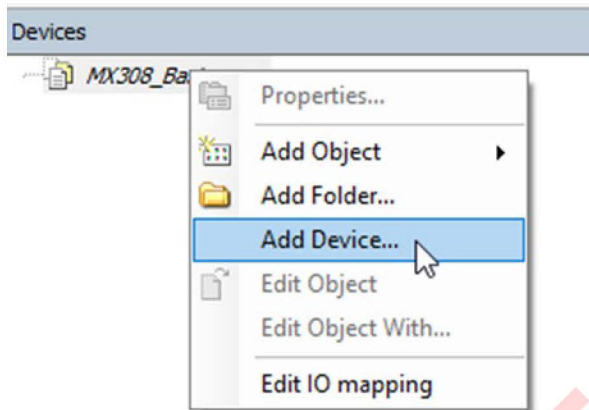
## Добавление контроллера в проект

Для добавления контроллера в проект необходимо выполнить следующие действия.

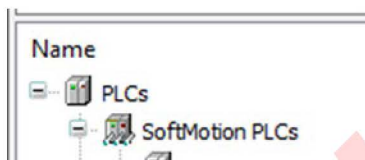
Щёлкните правой кнопкой мышки на названии проекта в левом верхнем углу проекта:



В появившемся меню выберите пункт **Add Device**:



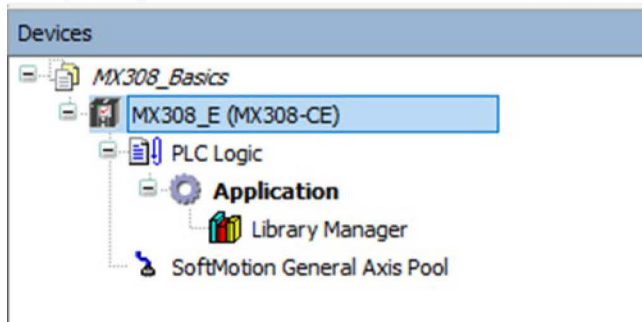
В открывшемся окне выберите раздел **PLC – SoftMotion PLC**:



Далее прокрутите вниз до пункта с контроллером MX300

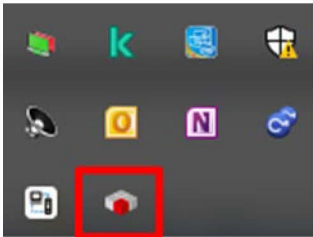
	CODESYS SoftMotion Win V3	3S - Smart Software Solutions GmbH	3.5.18.50	DEP
	CODESYS SoftMotion Win V3 x64	3S - Smart Software Solutions GmbH	3.5.18.50	DEP
	MX308-CE	Optimus Drive, Россия	3.5.15.40	Opt

выберите его и нажмите кнопку Add Device и закройте окно. В древе проекта появится устройство – контроллер MX300:

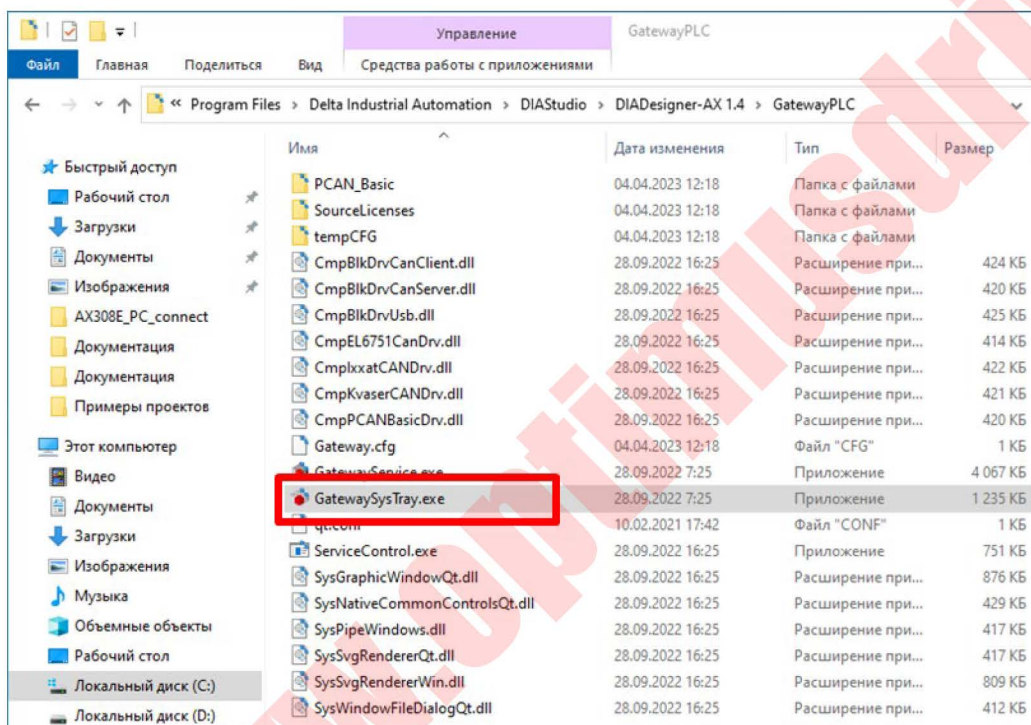


## Организация связи контроллеров типа MX300 и среды программирования. Загрузка программы. Онлайн режим

После установки среды программирования в Windows System Tray (правый нижний угол экрана) должна появиться иконка шлюза CODESYS Gateway V3:

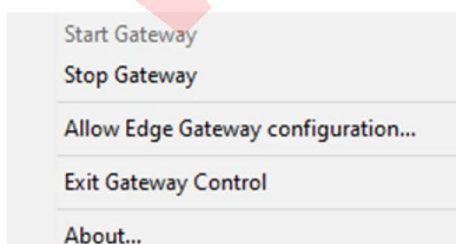


Если шлюз как приложение не запущен, то его можно запустить принудительно через исполнительный файл:



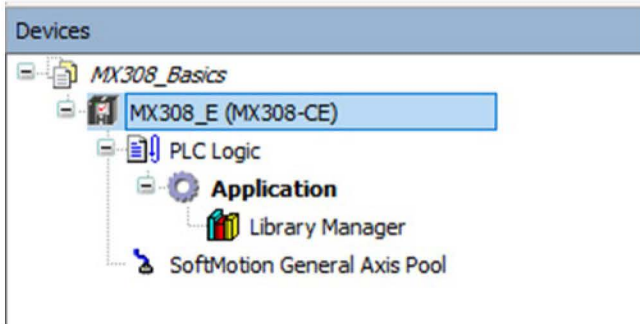
После этого должна появиться иконка.

Если щёлкнуть на иконке шлюза левой кнопки мышки, то откроется меню, позволяющее включить/выключить шлюз и закрыть приложение:

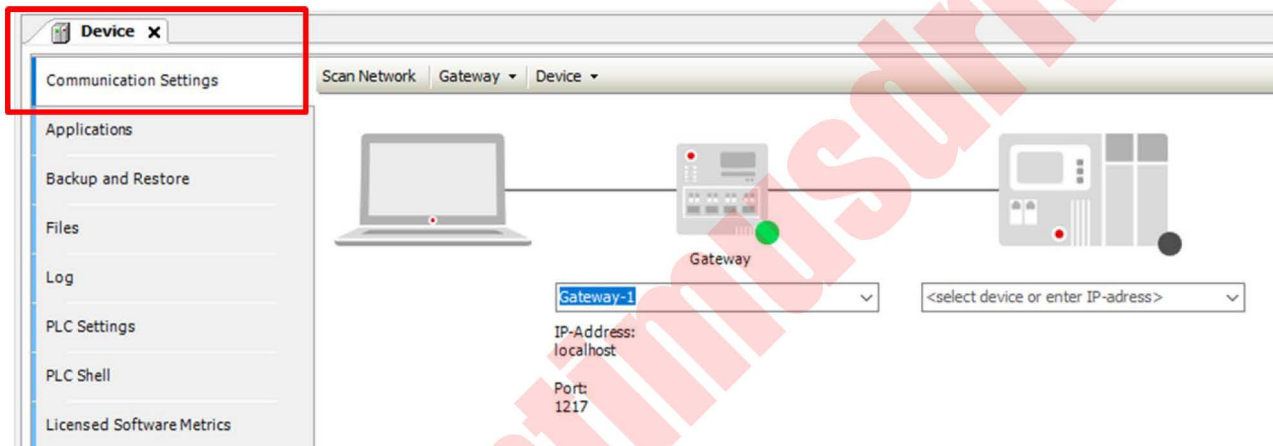


Шлюз используется для организации связи контроллера со средой программирования. Для связи можно использовать USB или Ethernet. Далее рассматриваются оба варианта подключения.

В древе проекта щёлкните дважды на названии контроллера:



Откроется вкладка **Device** пункт **Communication Settings**:



По индикатору зелёный/красный можно судить от том, включен шлюз или нет.



## Соединение через порт USB-C

USB драйвер устанавливается на ПК автоматически при инсталляции пэкиджа, см. соответствующий раздел данного Руководства.

Соедините ПК и контроллер через USB-C. В Диспетчере устройств не появляется при этом никаких записей. Драйвер работает через сетевые подключения, поэтому откройте Параметры сети и Интернет Windows – Настройки параметров адаптера:

### Дополнительные сетевые параметры



#### Настройка параметров адаптера

Просмотр сетевых адаптеров и изменение параметров подключения.

При подключении контроллера к ПК и правильно установленном драйвере должна появиться Неопознанная сеть данного вида:



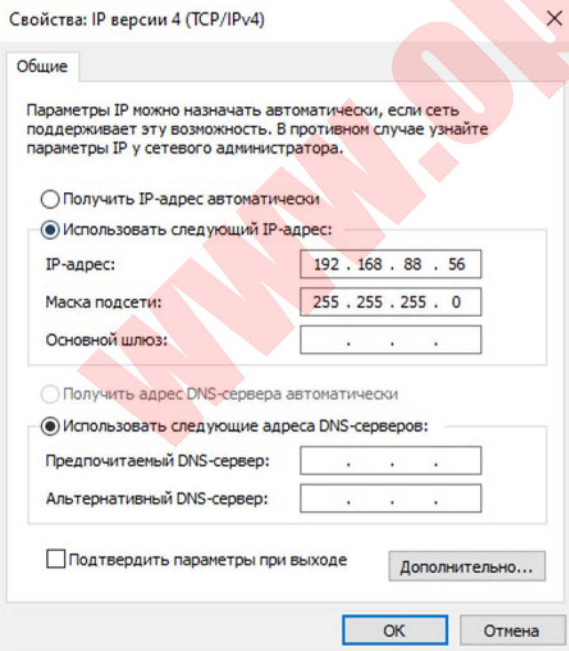
#### Ethernet 3

Неопознанная сеть

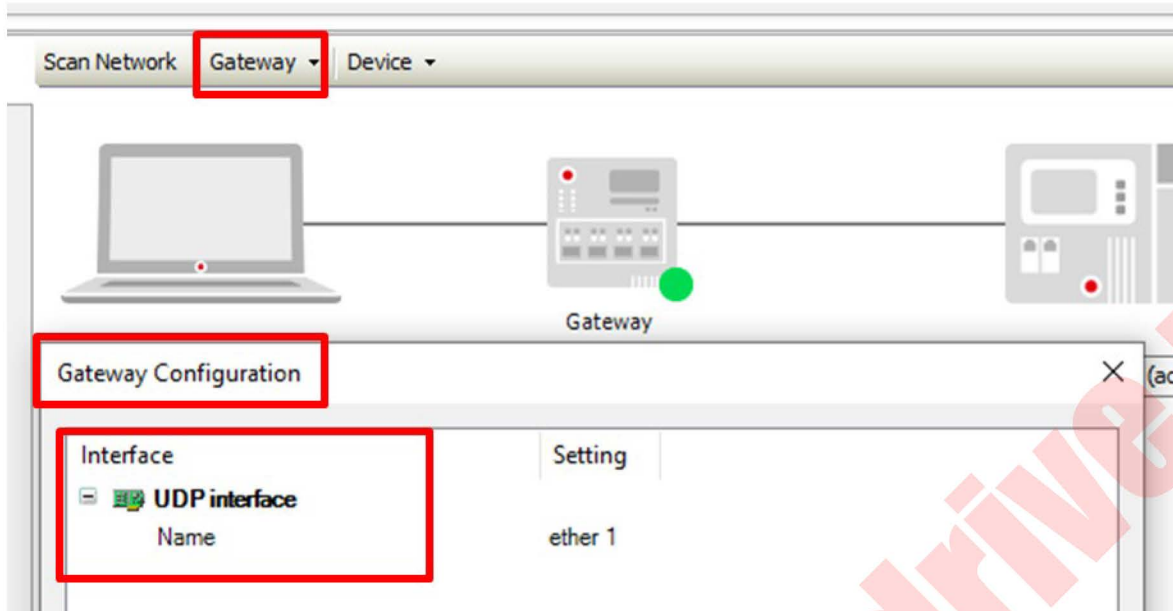
LeiSai USB Ethernet/RNDIS Gadget

При соединении по USB сетевой контроллера будет фиксированный: **192.168.88.88**

Поэтому адрес ПК в данном виртуальном адаптере должен быть в данной подсети, но не такой же:

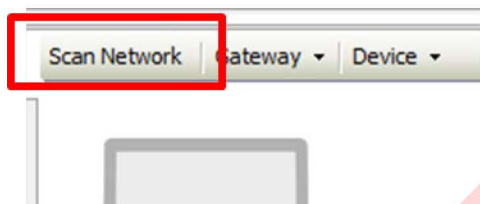


В общем случае для соединения достаточно драйвера по умолчанию:

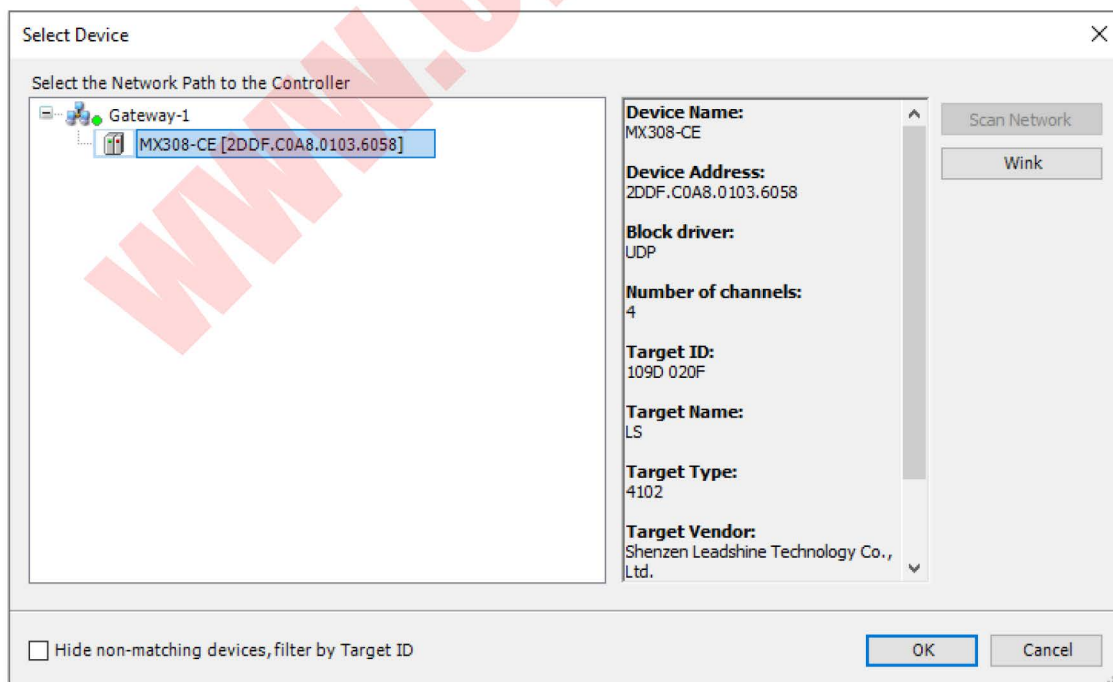


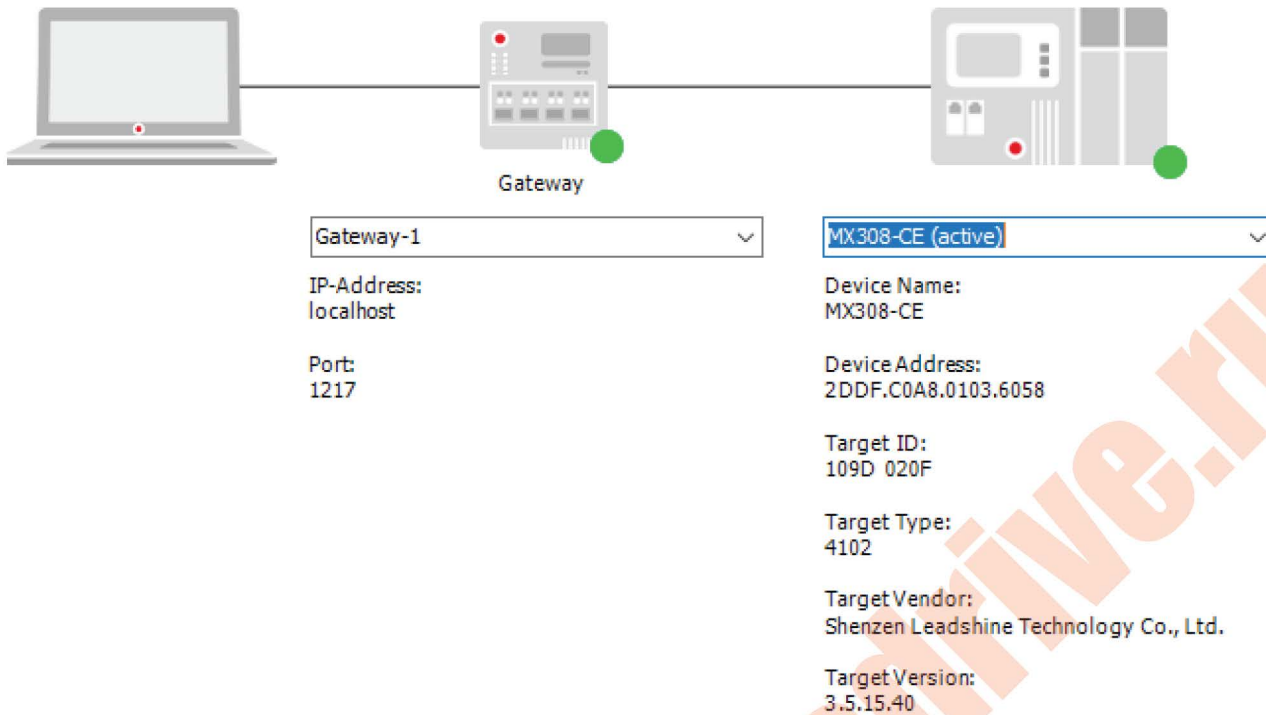
После этого можно приступить к установлению соединения среды программирования и контроллера.

В пункте **Communication Settings** нажмите иконку **Scan Network**:



Появится окно поиска и в итоге должен быть обнаружен контроллер:



**Внимание!**

Если соединение не устанавливается, то можно сбросить в контроллере настройки связи на заводские. Для этого необходимо быстро включить-выключить переключатель RUN-STOP на передней панели контроллера (под крышкой). На пятый раз должны включиться примерно на 1 секунду все светодиоды рядом с переключателем. После данной процедуры контроллер будет стоять на заводских настройках до следующего отключения питания.

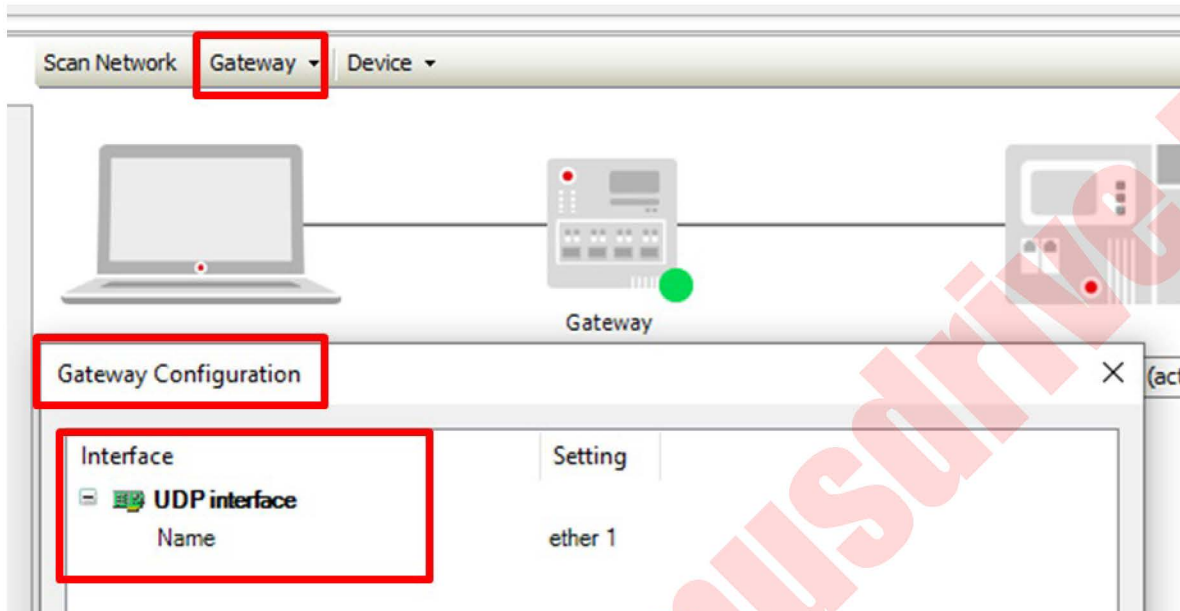


## Соединение через порт Ethernet

Для установления соединения посредством сети Ethernet соедините патчкордом порты контроллера и ПК напрямую или через неуправляемый коммутатор (свитч).

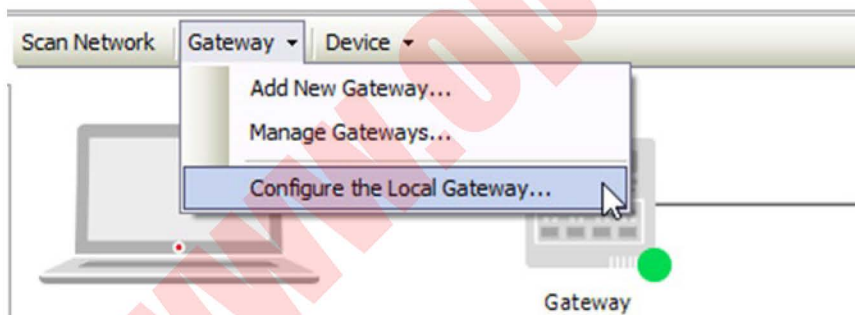
У контроллера IP адрес по умолчанию - **192.168.1.3**

В общем случае для соединения достаточно драйвера по умолчанию:

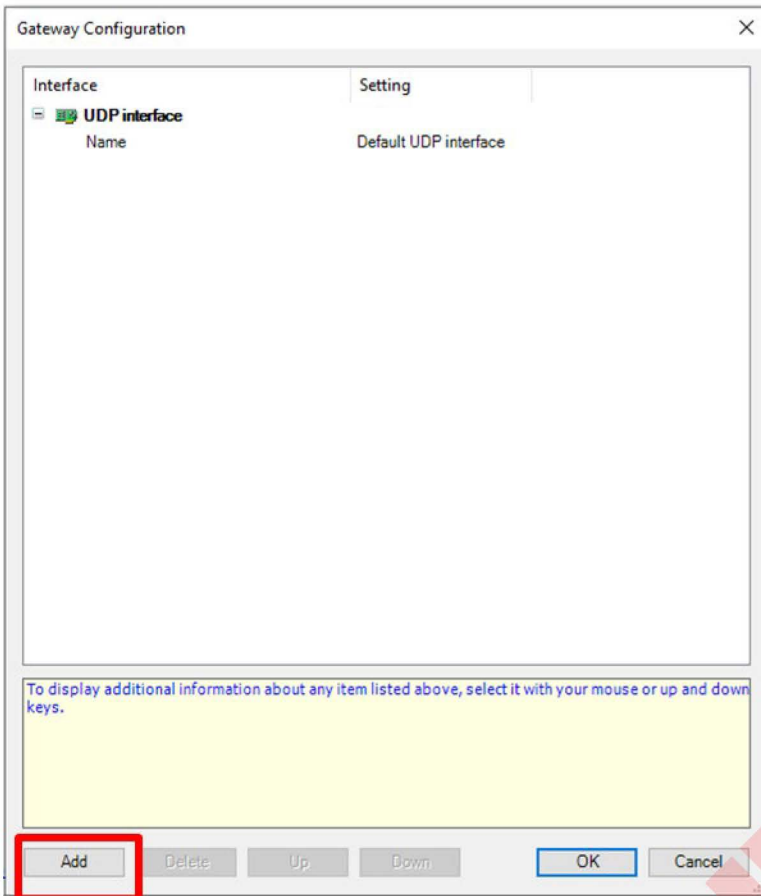


В случае соединения через несколько коммутаторов возможно потребуется создать явное TCP соединение.

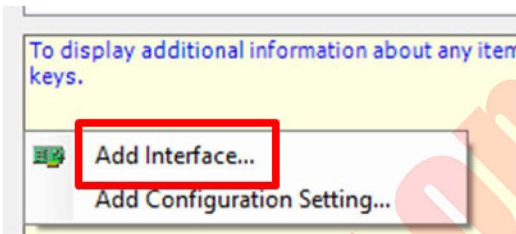
Для этого выберите пункт **Gateway – Configure the Local Gateway:**



Появится окно со списком каналов связи, которое в новом проекте будет пустым:

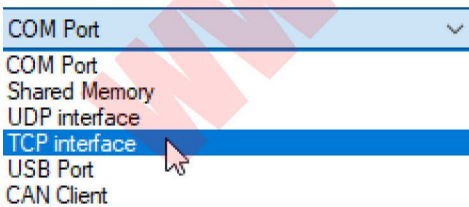


В левом нижнем углу нажмите кнопку **Add**, появится окно создания интерфейса:

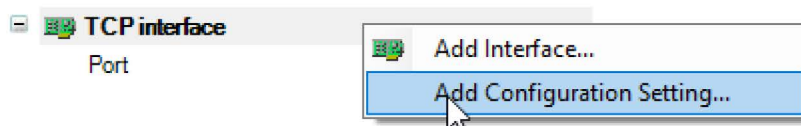


Выберите пункт **Add Interface**.

Появится ещё один пункт, где нужно выбрать **TCP interface**:



Щёлкните правой кнопкой мышки на надписи **TCP interface** и выберите пункт **Add Configuration Setting**:

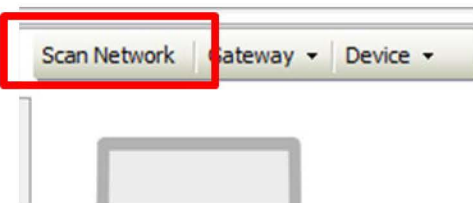


И выберите как минимум поле IP адреса. Введите IP адрес контроллера. При последующих нажатиях будут появляться следующие поля, которые можно не менять и не выводить.

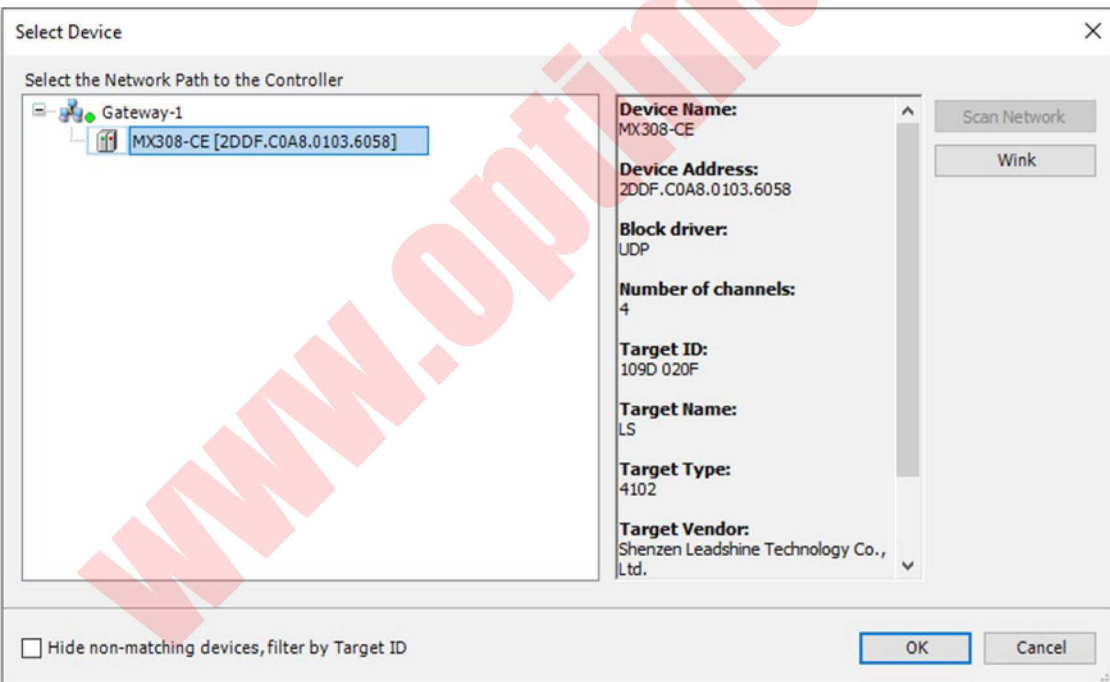
У контроллера IP адрес по умолчанию - **192.168.1.3**

Interface	Setting
<b>TCP interface</b>	
Port	11743
IP address	192.168.1.3
Inactivity timeout	60
Enable client	TRUE
Enable server	TRUE
Local access only	FALSE

После создания канала связи нажмите иконку **Scan Network**:



Появится окно с поиском и найденным контроллером:

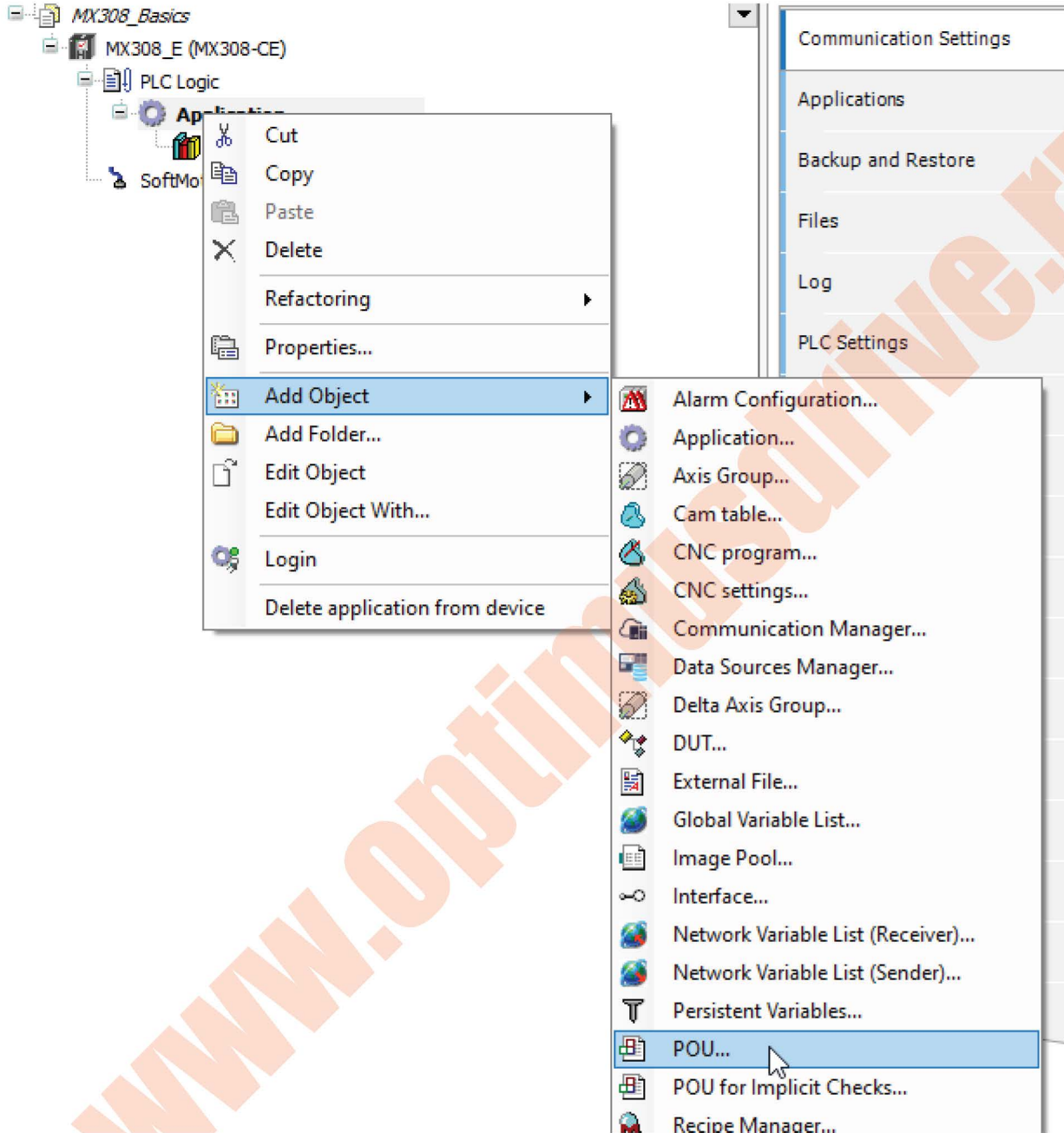


**Внимание!**

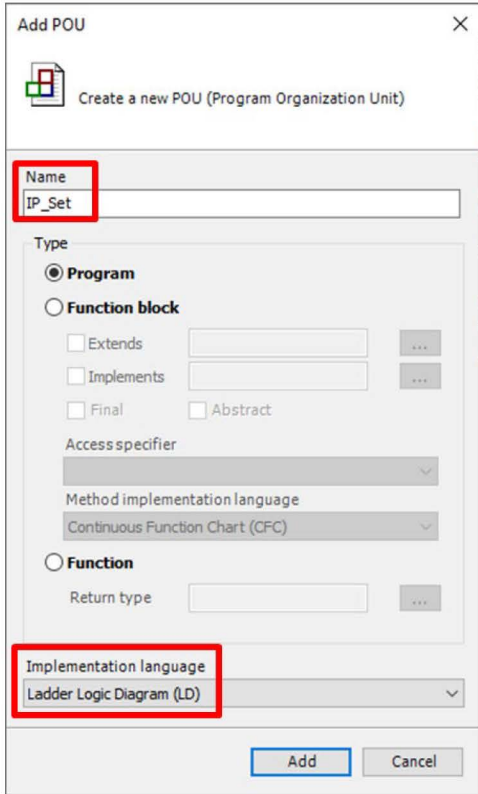
Если соединение не устанавливается, то можно сбросить в контроллере настройки связи на заводские. Для этого необходимо быстро включить-выключить переключатель RUN-STOP на передней панели контроллера (под крышкой). На пятый раз должны включиться примерно на 1 секунду все светодиоды рядом с переключателем. После данной процедуры контроллер будет стоять на заводских настройках до следующего отключения питания.

## Изменение IP адреса контроллера

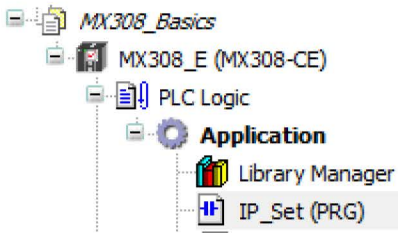
Создайте в проекте контроллера программную единицу (POU). Для этого щёлкните в древе проекта на пункте **Application** правой кнопкой мышки и в открывшемся меню выберите пункт **Add Object – POU**:



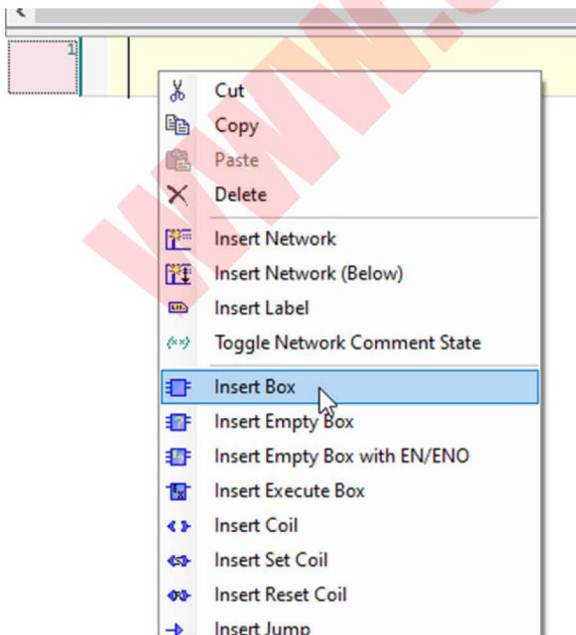
В открывшемся окне задайте название и язык программирования:



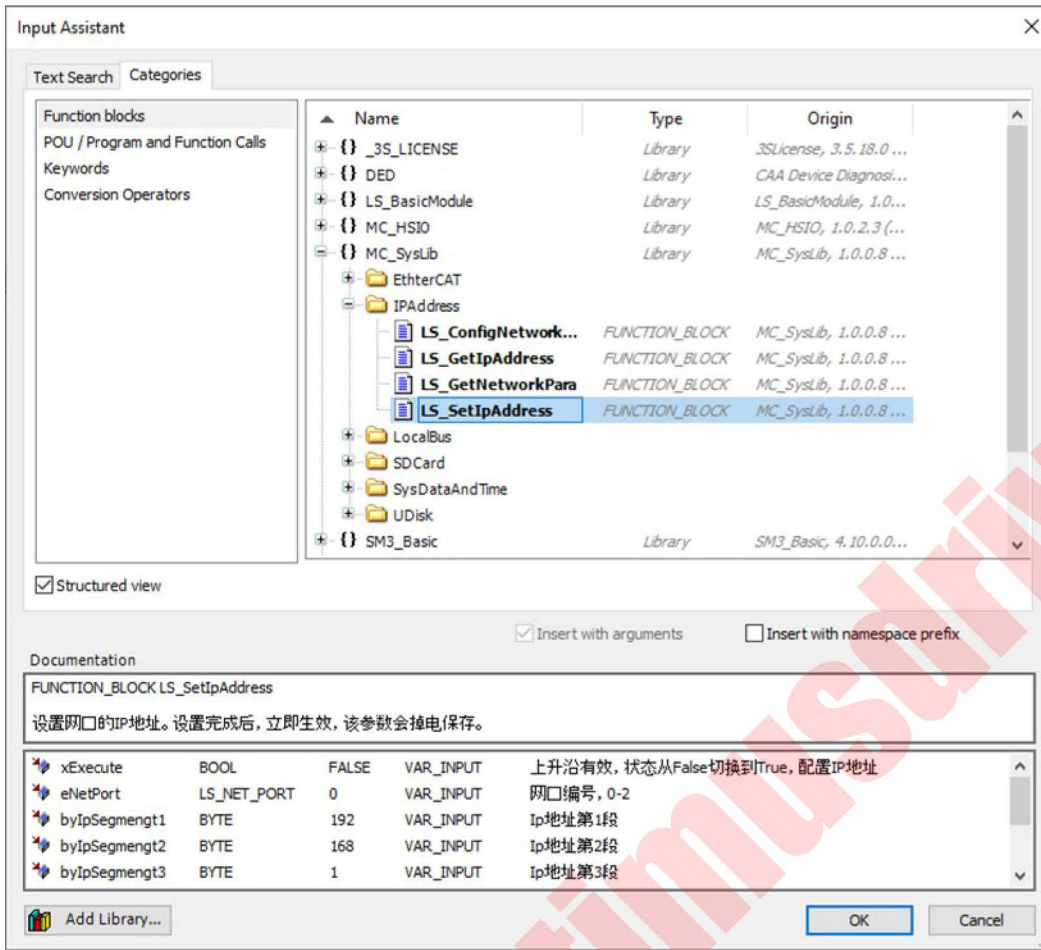
Нажмите кнопку **Add**. В древе проекта появится POU IP\_Set



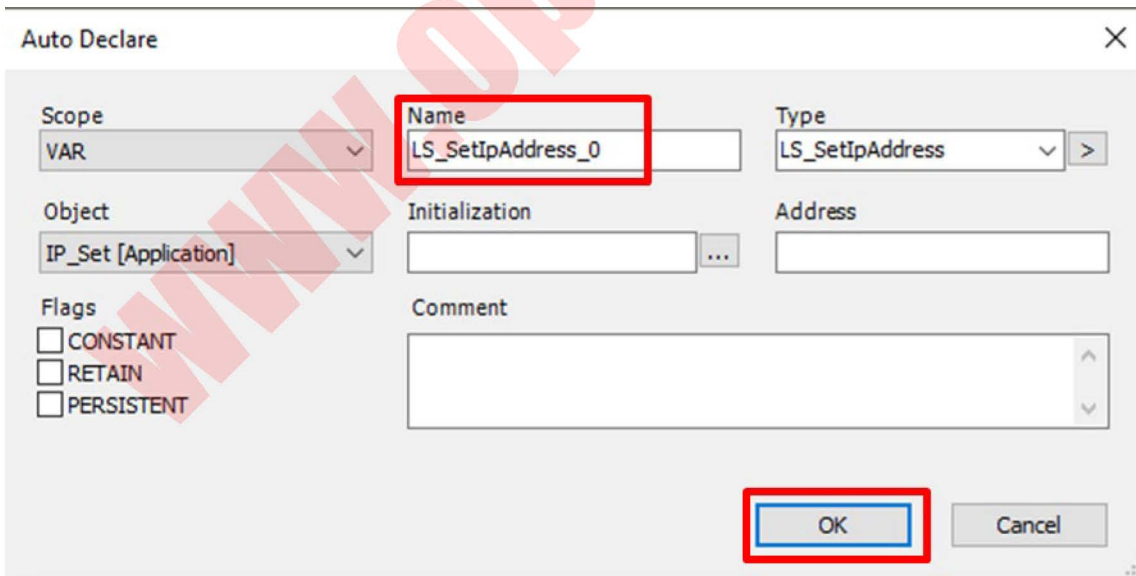
В окне созданного POU в поле ввода кода нажмите правой кнопкой мышки и выберите пункт **Insert Box**:



Откроется Мастер выбора команд. Нужно выбрать команду LS\_SetIpAddress (библиотека MC\_SysLib)

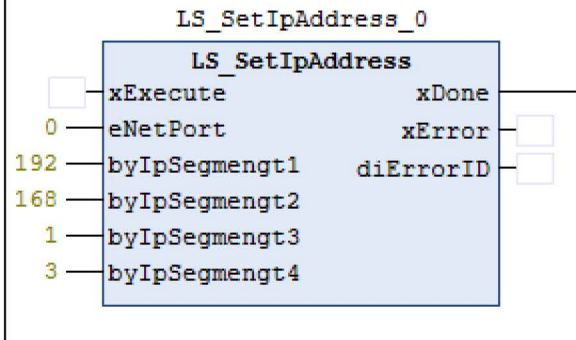


Задайте название экземпляра ФБ и нажмите **OK**:



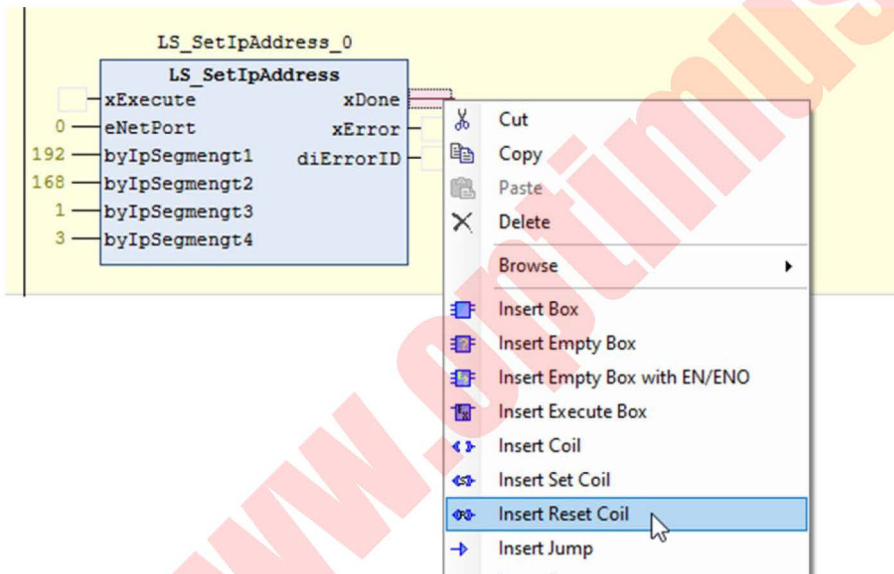
Появится переменная в таблице, а команда на строке кода:

Scope	Name	Address	Data type	Initialization	Comment	Attributes
1	VAR	LS_SetIpAddress_0	LS_SetIpAddress			

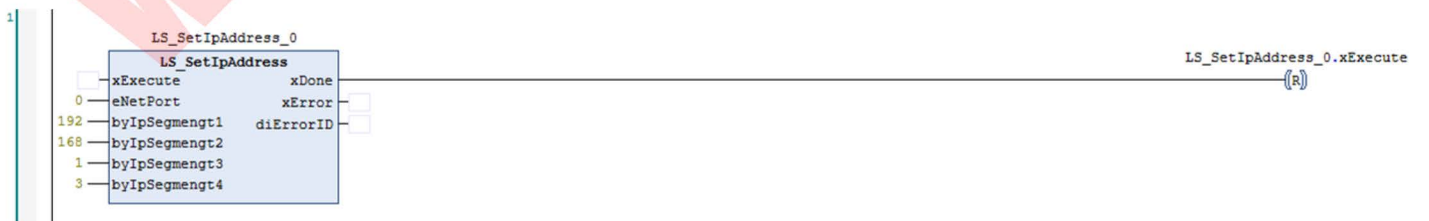


Введите нужный IP адрес (в примере вверху задан 192.168.1.3)

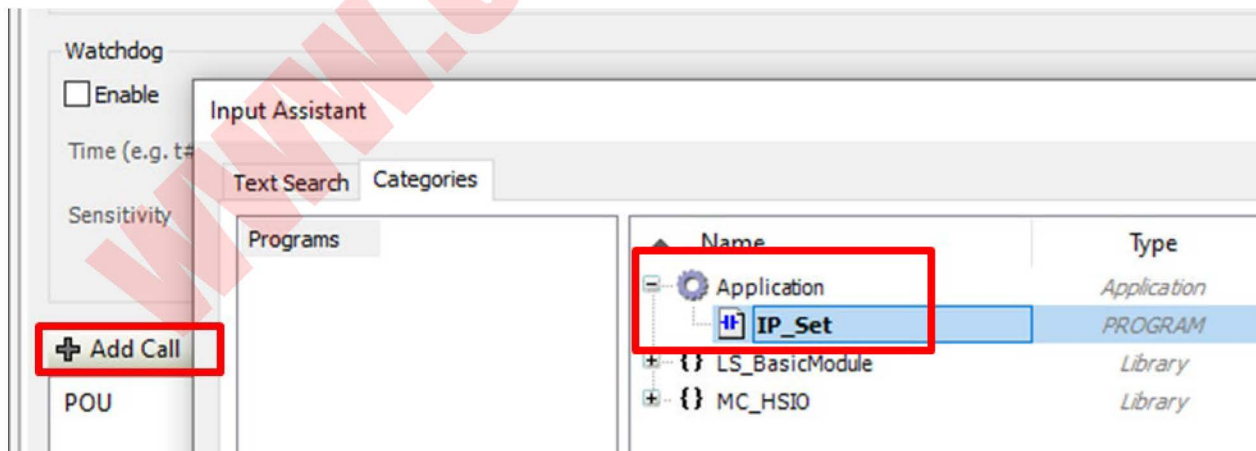
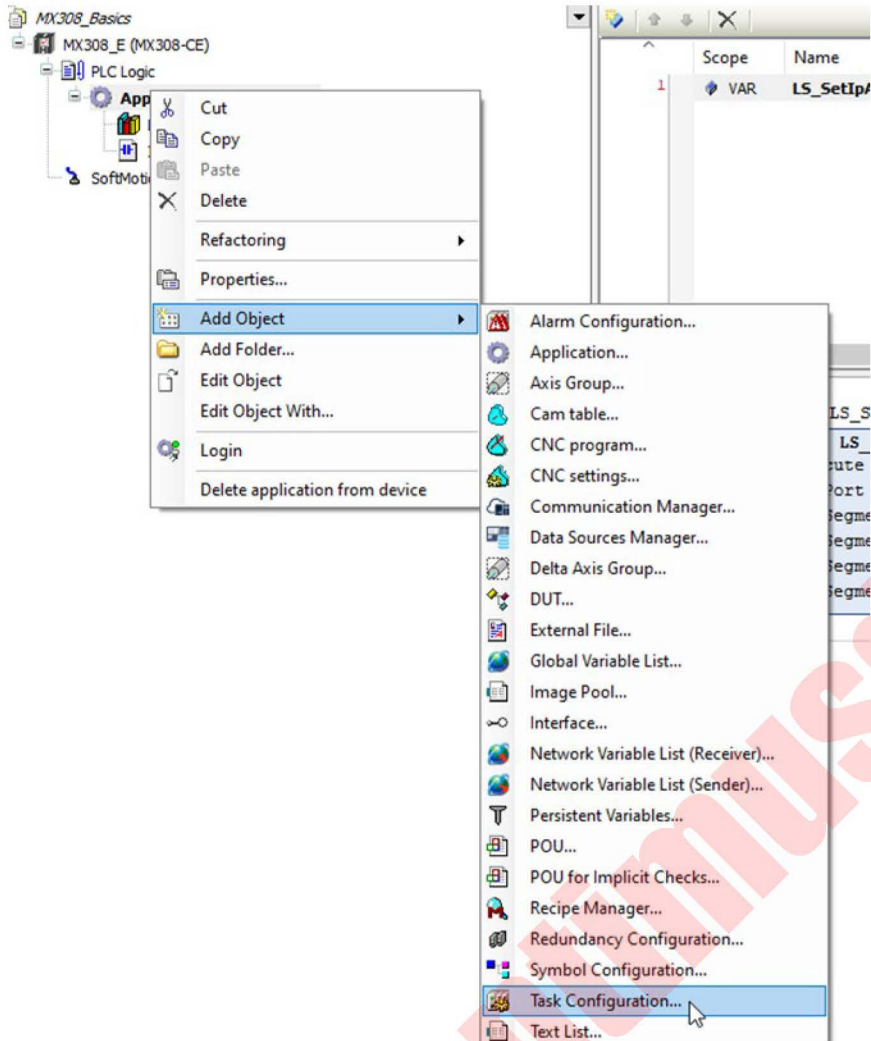
На ножку xDone установите катушка типа RST со следующей переменной LS\_SetIpAddress\_0.xExecute:



В итоге получится код следующего вида:

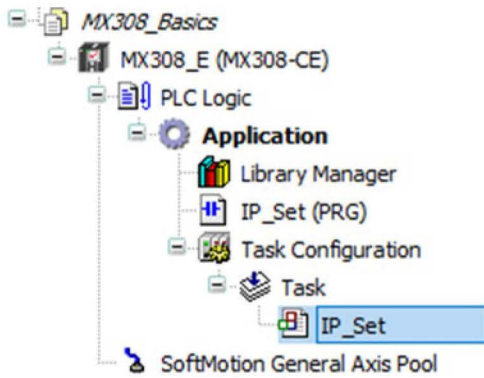


Далее необходимо добавить в проект Менеджер задач и добавить в него POU IP\_Set:





В древе проекта появится соответствующий пункт:

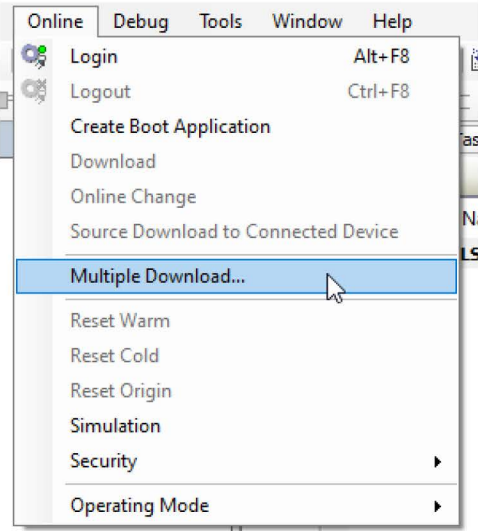


**Внимание!** Для активации нового IP адреса необходимо загрузить программу в контроллер и кратковременно перевести его в RUN.

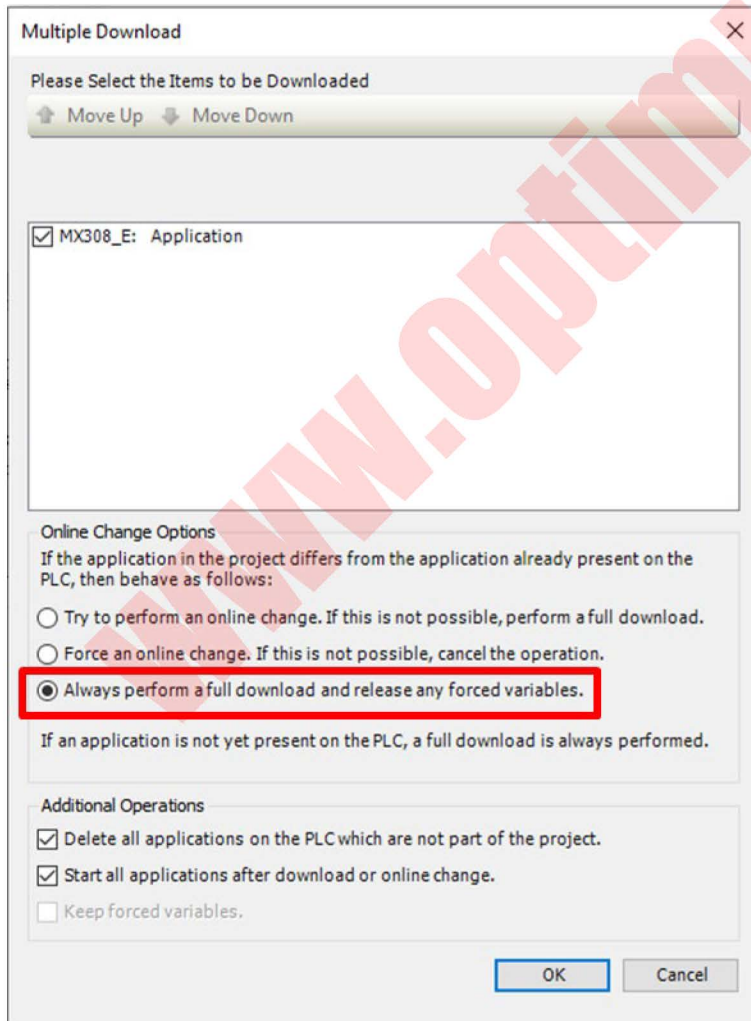
*См. Пример OD Samples – Nets - IpAddrChg\_SD.project*

## Загрузка проекта в контроллер и вход в режим онлайн

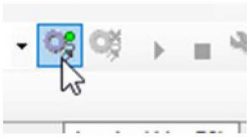
Для загрузки проекта в контроллер необходимо установить связь ПК – контроллер (см. выше), выбрать в меню Online выбрать пункт Multiple Download:



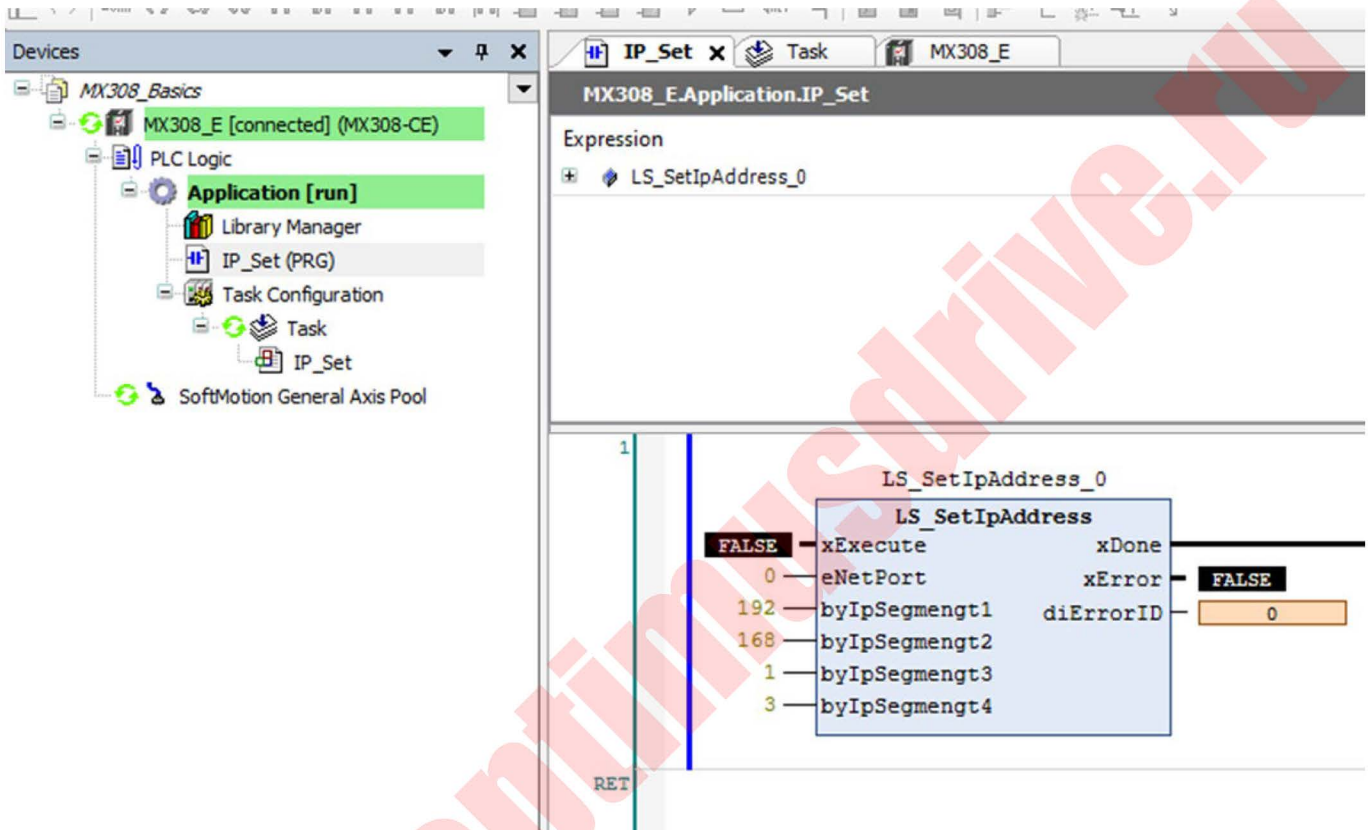
В открывшемся окне выберите вариант полной загрузки и нажмите **ОК**:



Для входа в режим онлайн нажмите кнопку:



Программа подсветится состоянием объектов:



Для выхода из режима онлайн нажмите кнопку с красным крестиком:

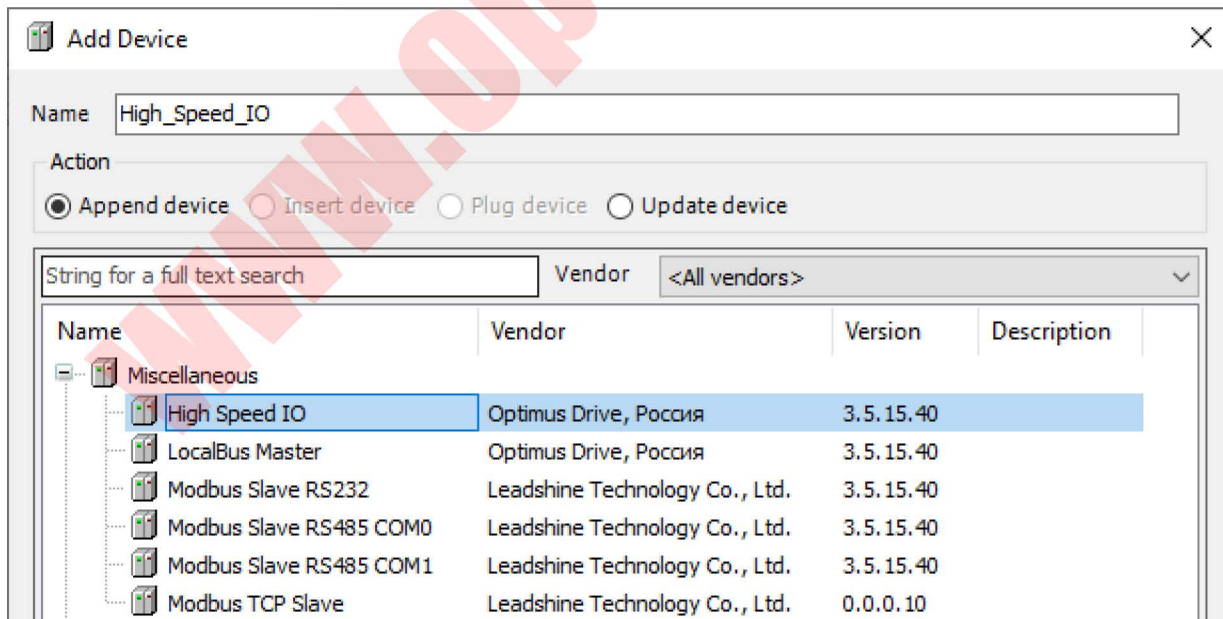


## Использование встроенных входов-выходов контроллера в обычном режиме

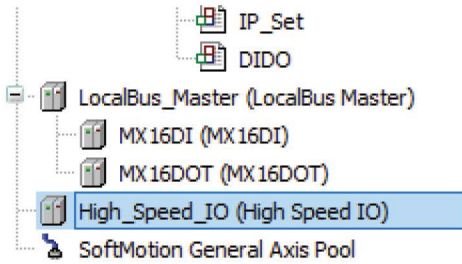
Для использования входов-выходов на ЦПУ в обычном режиме (не в импульсном) достаточно добавить в проект объект типа **High Speed IO** (идёт в составе пэкиджа, см. выше) для чего щёлкните правой кнопкой мышки на контроллере в древе проекта и в отрывшемся меню выберите пункт **Add Device**:



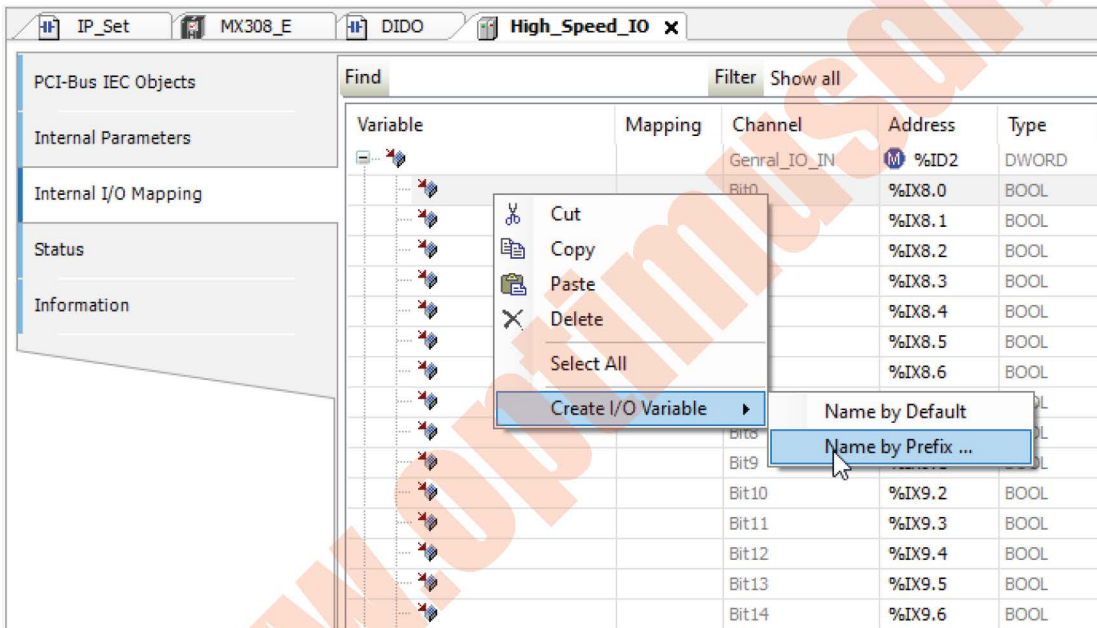
В появившемся окне выберите папку **Miscellaneous** и пункт **High Speed IO** и нажмите кнопку **Add Device**:



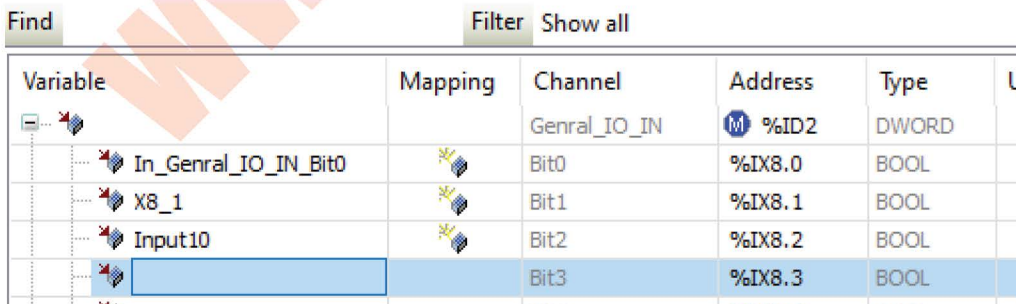
В древе проекта появится пункт **High Speed IO**:



Щёлкните на пункте High Speed IO дважды левой кнопкой мышки, откроется вкладка параметров входов-выходов на ЦПУ. Выберите в ней пункт Internal I/O Mapping, в которой будут видны физические адреса входов и выходов. Здесь же можно задать теги для них. Для этого разверните список и щёлкните правой кнопкой мышки на самом верхнем входе. В открывшемся окне выберите пункт Create I/O variables и, например, Name by Prefix.



Заполните колонку тегов путём подставки по префиксу, или просто вручную своих названий:



Теги для выходов заполняются аналогичным образом.

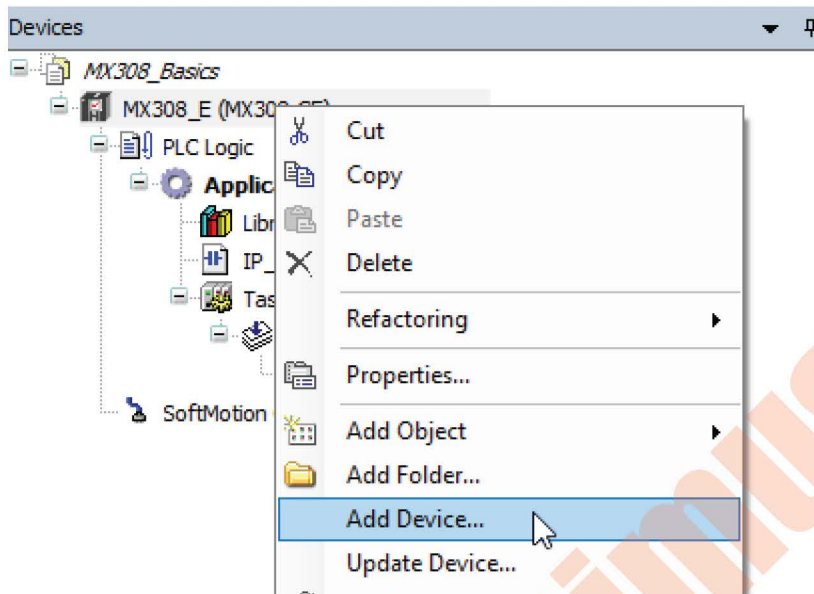
В программе состояние входов можно считывать как по тегам, так и по физическим адресам.

## Добавление в проект модулей расширения

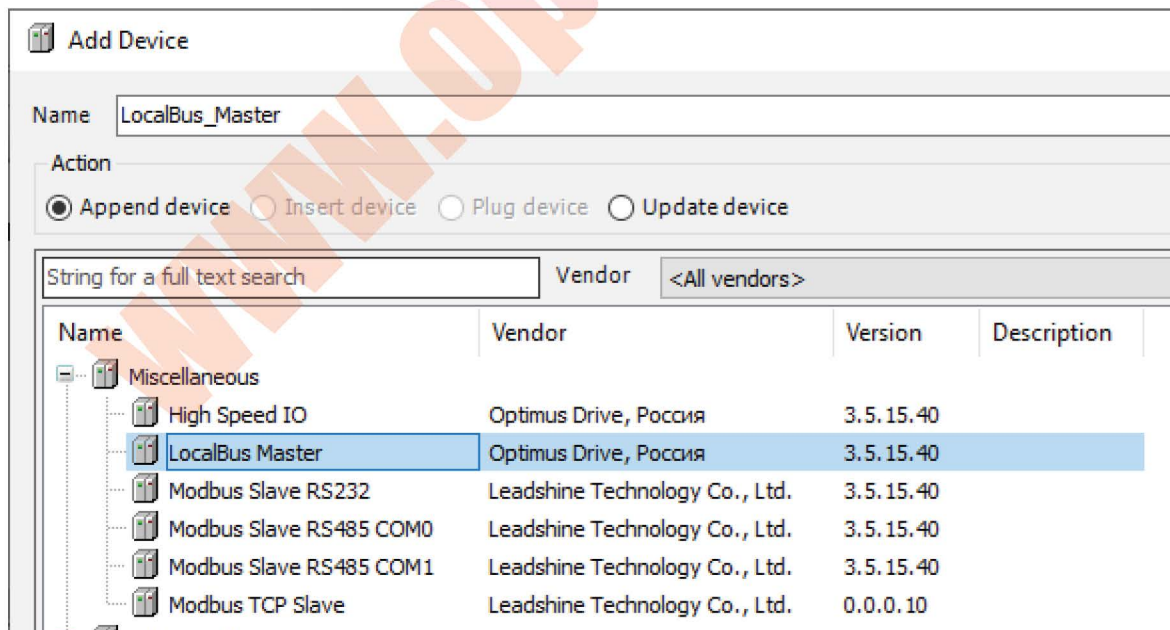
Контроллеры серии MX300 могут расширяться модулями дискретного и аналогового ввода-вывода. Всего к контроллеру по внутренней шине могут быть подключены до 32-х модулей расширения.

Файлы с описанием устройств (XML) устанавливаются в составе пэкиджа (см. выше).

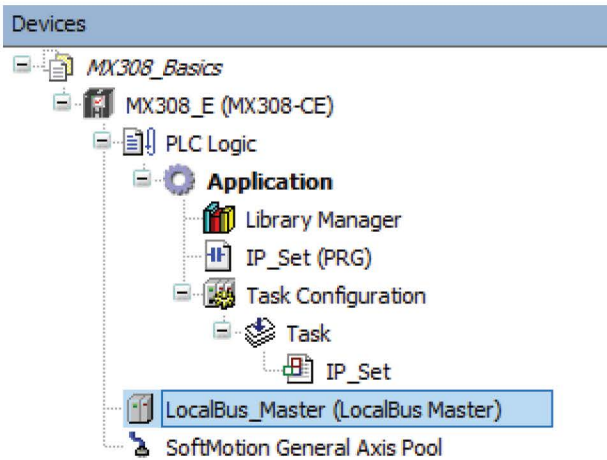
Для добавления модуля расширения в проект щёлкните правой кнопкой мышки на контроллере в древе проекта и в отрывшемся меню выберите пункт **Add Device**:



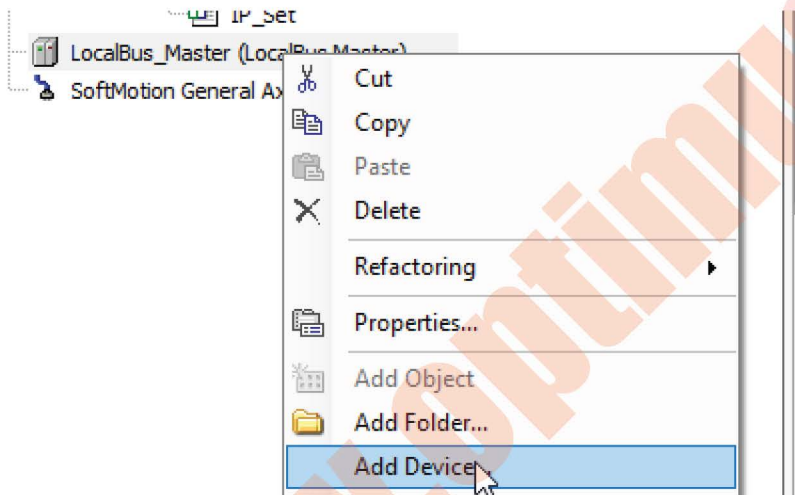
В появившемся окне выберите папку **Miscellaneous** и пункт **LocalBus Master** и нажмите кнопку **Add Device**:



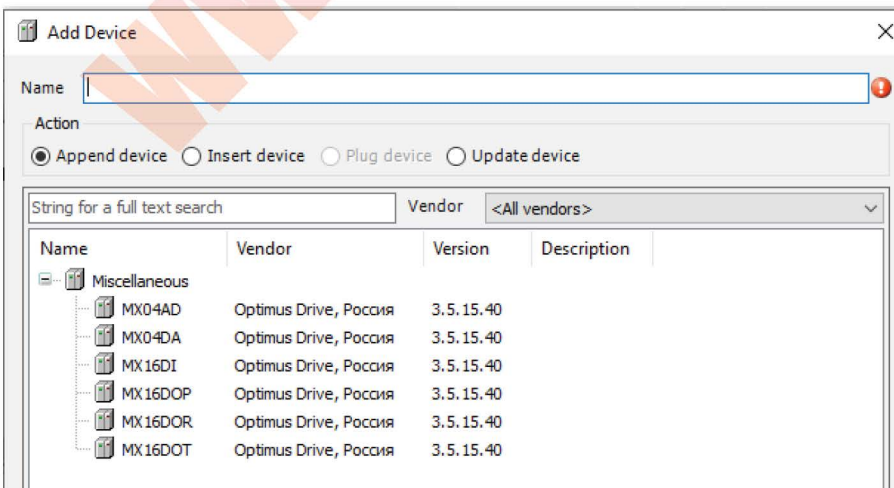
В древе проекта появится пункт **LocalBus Master**:



Щёлкните правой кнопкой мышки на пункте LocalBus Master и в отрывшемся меню выберите пункт **Add Device**:

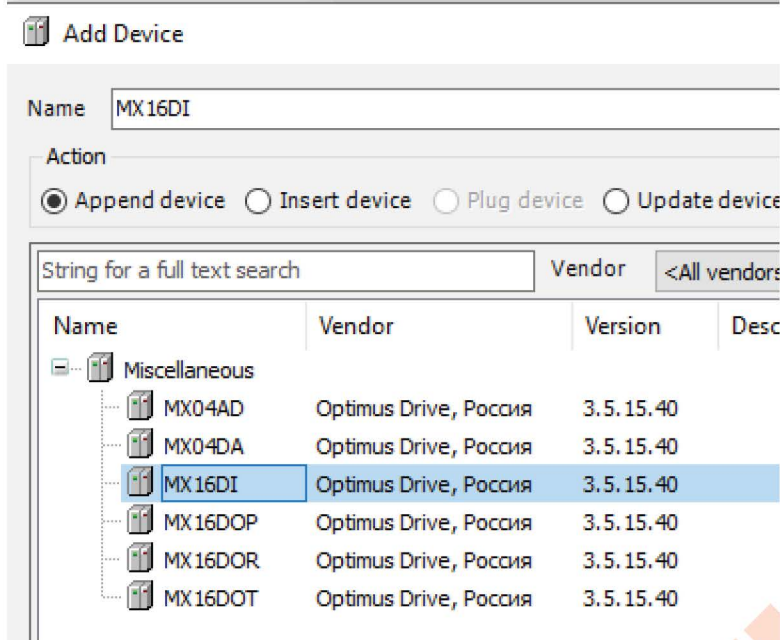


В открывшемся окне выберите нужный модуль:

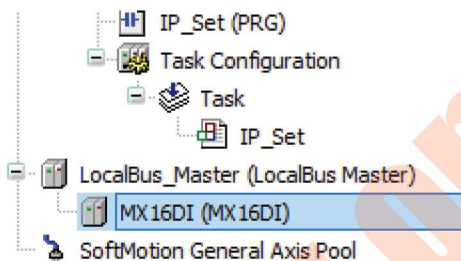


## Работа с модулем дискретных входов MX16DI

Добавьте в проект по вышеприведённой последовательности модуль MX16DI:

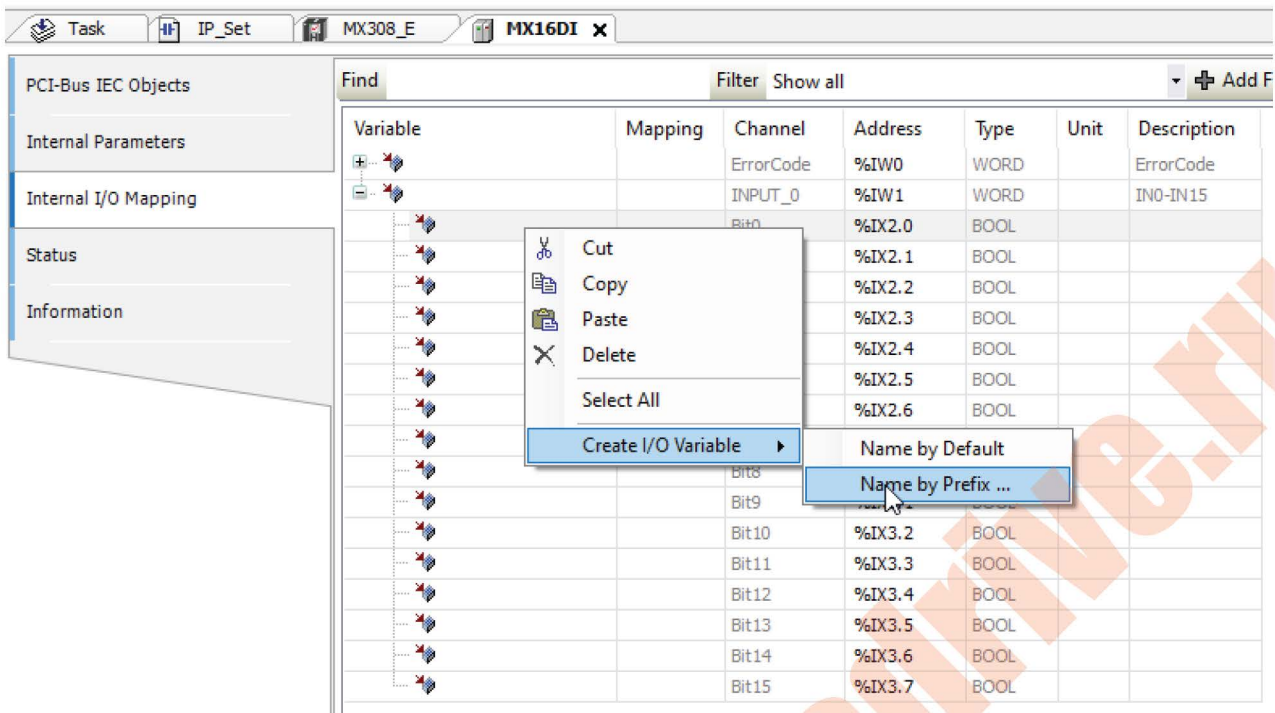


В проекте появится пункт:

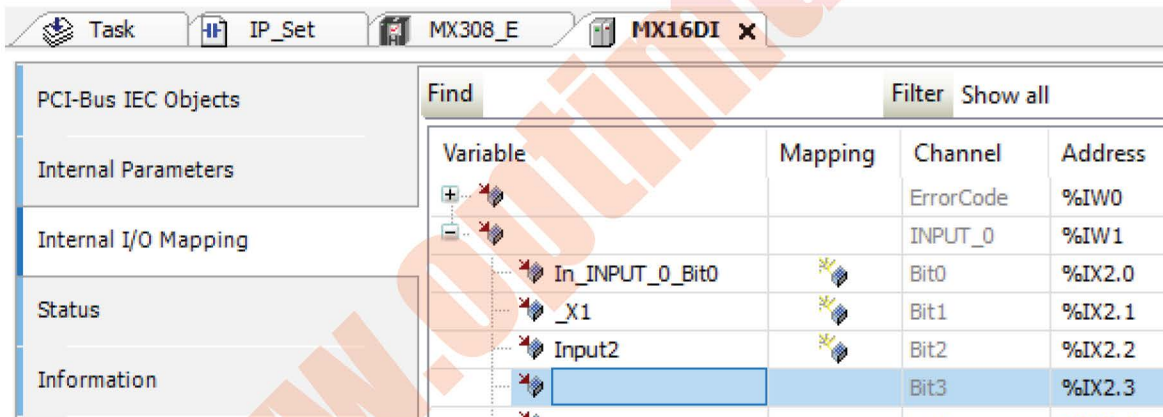


Щёлкните на пункте MX16DI дважды левой кнопкой мышки, откроется вкладка параметров модуля. Выберите в ней пункт Internal I/O Mapping, в которой будут видны физические адреса входов и можно задать теги для входов. Для этого разверните список и щёлкните правой кнопкой мышки на самом верхнем входе. В открывшемся окне выберите пункт Create I/O variables и, например, Name by Prefix.

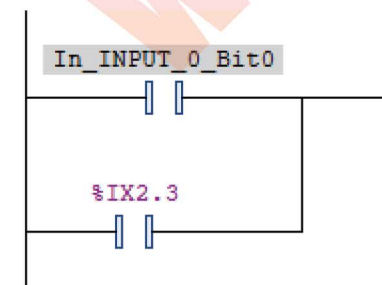




Заполните колонку тегов путём подставки по префиксу, или просто вручную своих названий:

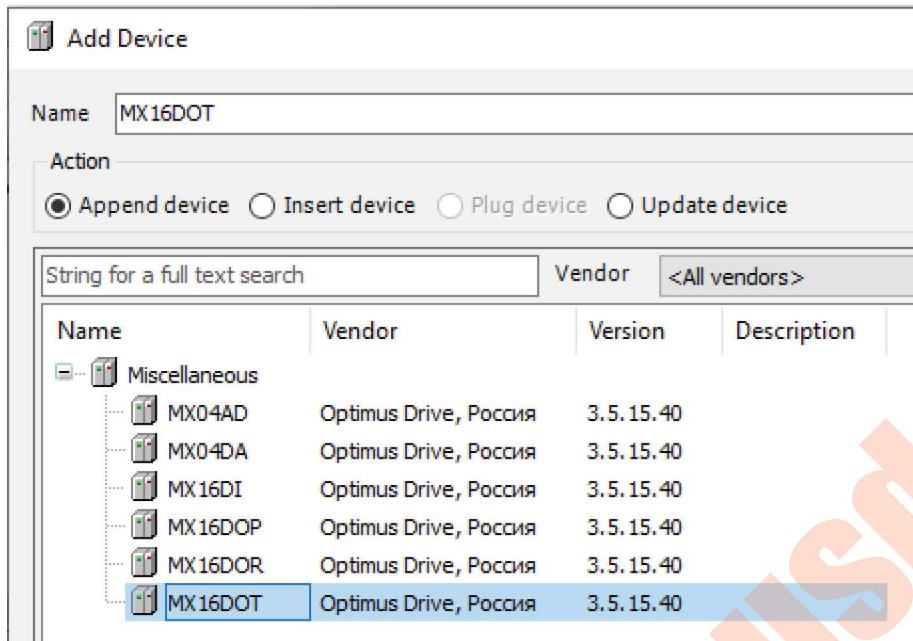


В программе состояние входов можно считывать как по тегам, так и по физическим адресам.

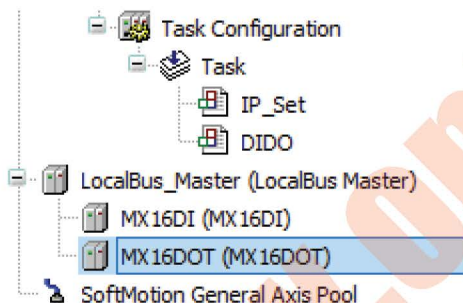


## Работа с модулями дискретных выходов MX16DOT/MX16DOP/MX16DOR

Добавьте в проект по вышеприведённой последовательности модуль MX16DOT:  
(модули MX16DOP и MX16DOR настраиваются аналогично)



В проекте появится пункт:



Щёлкните на пункте MX16DOT дважды левой кнопкой мышки, откроется вкладка параметров модуля. Выберите в ней пункт Internal I/O Mapping, в которой будут видны физические адреса выходов и можно задать теги для выходов. Для этого разверните список и щёлкните правой кнопкой мышки на самом верхнем выходе. В открывшемся окне выберите пункт Create I/O variables и, например, Name by Prefix.

IP\_Set | MX308\_E | MX16DI | DIDO | MX16DOT x

PCI-Bus IEC Objects  
Internal Parameters  
Internal I/O Mapping  
Status  
Information

Find Filter Show all

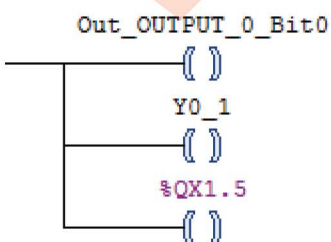
Variable	Mapping	Channel	Address	Type	Unit
		ErrorCode	%IW2	WORD	
		OUTPUT_0	%QW0	WORD	
		Bit0	%QX0.0	BOOL	
			%QX0.1	BOOL	
			%QX0.2	BOOL	
			%QX0.3	BOOL	
			%QX0.4	BOOL	
			%QX0.5	BOOL	
			%QX0.6	BOOL	
		Bit8			
		Bit9			
		Bit10	%QX1.2	BOOL	
		Bit11	%QX1.3	BOOL	
		Bit12	%QX1.4	BOOL	
		Bit13	%QX1.5	BOOL	
		Bit14	%QX1.6	BOOL	
		Bit15	%QX1.7	BOOL	

Context menu: Cut, Copy, Paste, Delete, Select All, Create I/O Variable (Name by Default, Name by Prefix)

Заполните колонку тегов путём подставки по префиксу, или просто вручную своих названий:

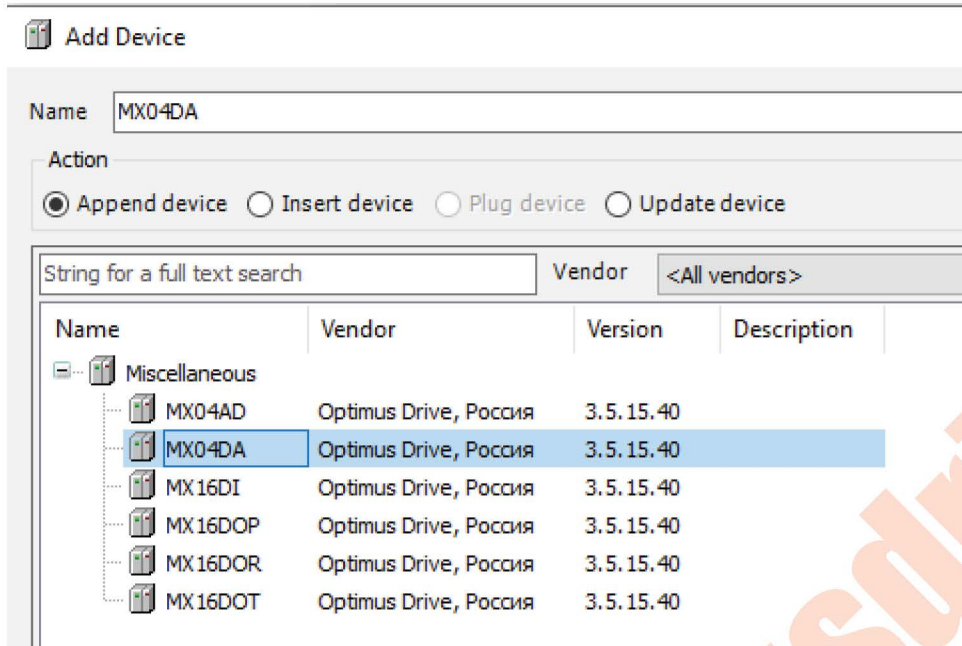
Variable	Mapping	Channel	Address	Type	Unit
		ErrorCode	%IW2	WORD	
		OUTPUT_0	%QW0	WORD	
Out_OUTPUT_0_Bit0		Bit0	%QX0.0	BOOL	
Y0_1		Bit1	%QX0.1	BOOL	
Output2		Bit2	%QX0.2	BOOL	
		Bit3	%QX0.3	BOOL	
		Bit4	%QX0.4	BOOL	

В программе состояние входов можно считывать как по тегам, так и по физическим адресам.

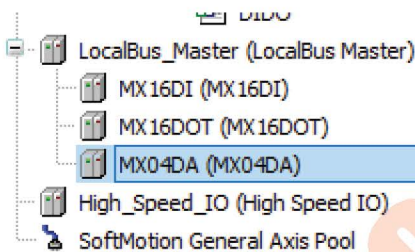


## Работа с модулем аналоговых выходов MX04DA

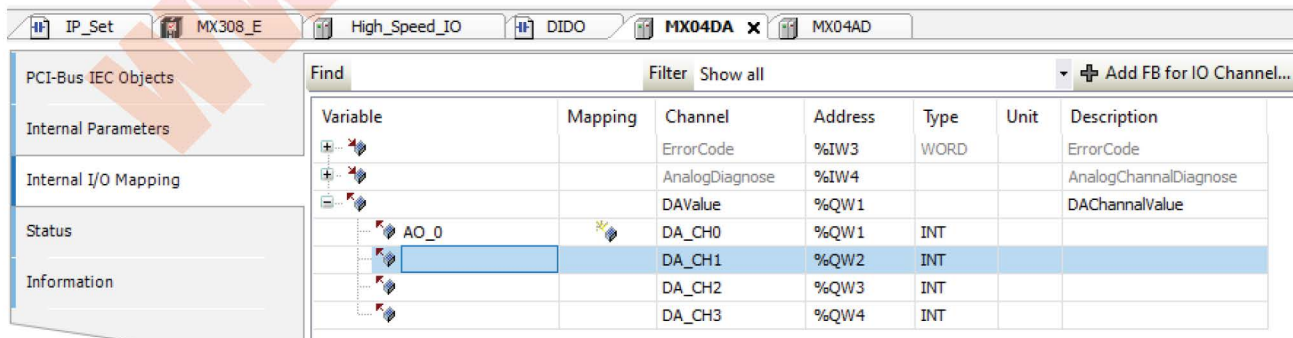
Добавьте в проект по вышеприведённой последовательности модуль MX04DA:



В проекте появится пункт:



Щёлкните на пункте MX04DA дважды левой кнопкой мышки, откроется вкладка параметров модуля. Выберите в ней пункт Internal I/O Mapping, в которой будут видны физические адреса выходов и можно задать теги для выходов. Для этого разверните список и введите нужное название тега.



В программе к аналоговым выходам можно обращаться как по тегам, так и по физическому адресу.

Для выбора режима работы аналогового выхода необходимо открыть вкладку **Internal Parameters**. Выберите раздел типа **DA0 Config**. Для каждого канала DA0 – DA3 предусмотрен свой раздел. В данном разделе выберите пункт **Data Buffer**.

Parameter	Type	Value	Default Value	Unit
LocalBusSlave Info				
SlaveAddr	BYTE	0	0	
SlaveNodeID	dword	1627390469	1627390469	
DA0 config				
Index	UINT	16#8000	16#8000	
SubIndex	UINT	16#01	16#01	
<b>Data_Buffer</b>	UINT	0	0	
Data_Size	UINT	1	1	

В пункт **Data Buffer** вносится номер режима работы в соответствии с нижеприведённой таблицей:

Значение в Data_Buffer	Тип аналогового сигнала	Диапазон цифровой шкалы
0	0~5V	0~32000
1	1~5V,	0~32000
2	-5~5V,	-32000~32000
3	0~10V	0~32000
4	-10~+10V	-32000~32000
5	0~20mA	0~32000
6	4~20mA	0~32000

Далее, в разделе типа **DA0\_en** в пункте **Data Buffer** для разрешения работы канала нужно выставить 1, а для запрета 0. По умолчанию работа каналов разрешена.

Parameter	Type	Value	Default Value	Unit
DA0 en				
Index	UINT	16#8001	16#8001	
SubIndex	UINT	16#01	16#01	
<b>Data_Buffer</b>	UINT	1	1	
Data_Size	UINT	1	1	

В разделе типа **DA0 state when link lost** в пункте **Data Buffer** можно установить реакцию канала на потерю связи:

Parameter	Type	Value	Default Value	Unit
DA0 state when link lost				
Index	UINT	16#8002	16#8002	
SubIndex	UINT	16#01	16#01	
<b>Data_Buffer</b>	UINT	0	0	
Data_Size	UINT	1	1	

Значение в Data_Buffer	Реакция канала
0	Сохранять текущее значение
1	Сброс на 0
2	Выдача заранее предустановленного значения

Предустановленное значение выставляется в разделе типа **DA0 value when link lost**:

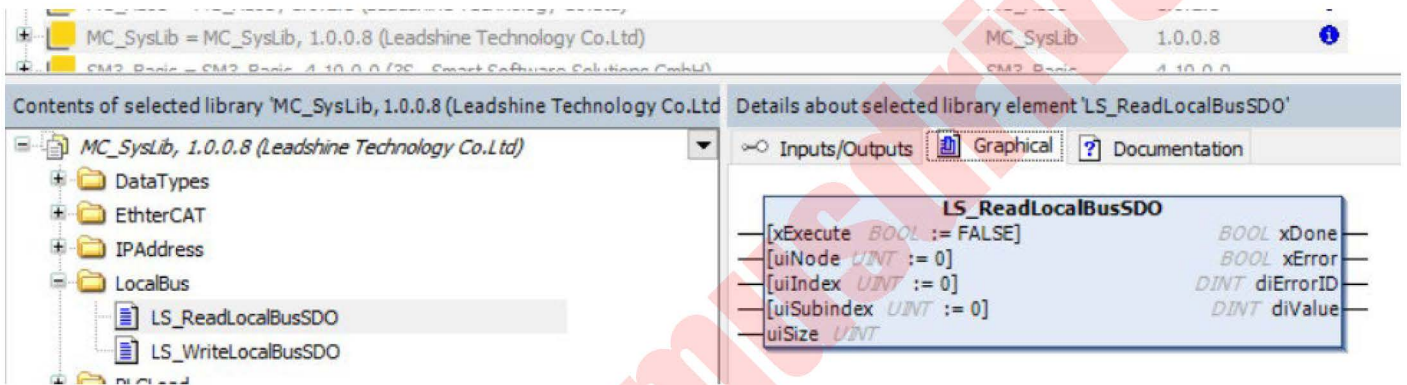
DA0 value when link lost				
Index	UINT	16#8003	16#8003	
SubIndex	UINT	16#01	16#01	
Data_Buffer	UINT	0	0	
Data_Size	UINT	2	2	

Например, если для канала выбран режим 0~10 VDC, то при установке в Data\_Buffer значения 27200 на аналоговом выходе при пропадании связи с модулем будет 8.50 VDC.

Модуль можно конфигурировать и из программы. Для этого используются команды типа:

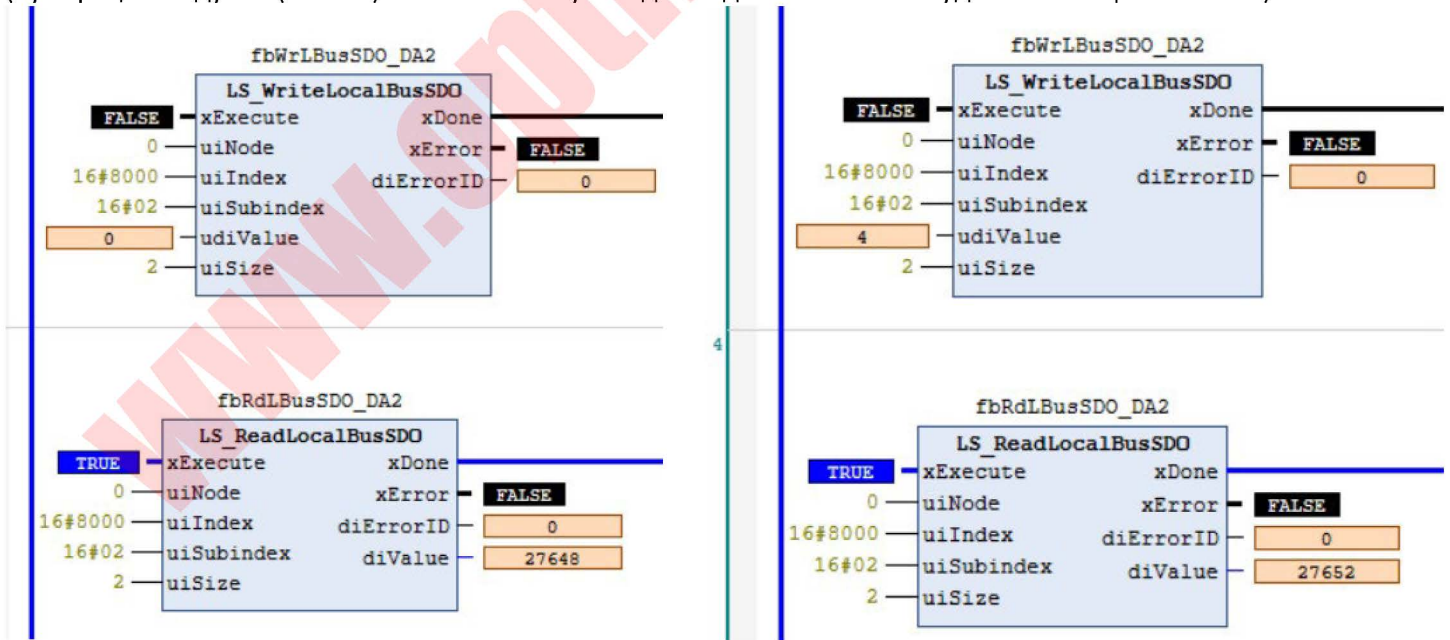
**LS\_ReadLocalBusSDO**

**LS\_WriteLocalBusSDO**



Например, запись режима 0 и чтение обратно для проверки:

(нумерация модулей (uiNode) начинается с нуля и далее до 31 по степени удаления от ЦПУ на шине)

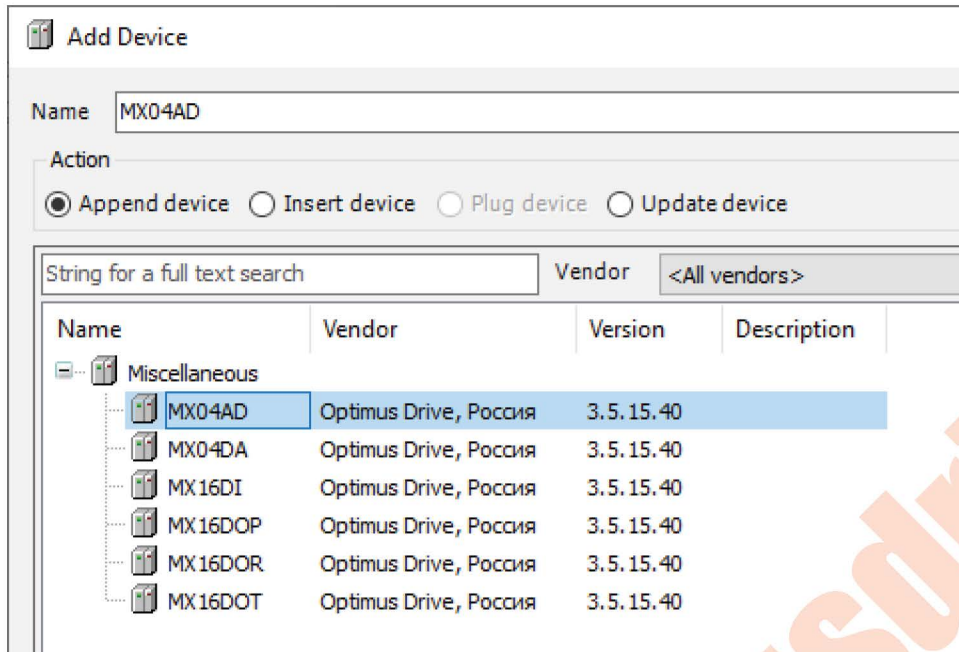


Data\_Buffer = 0 - на записи, соответствует 27648 - на чтении

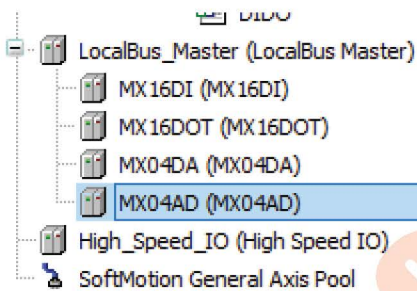
Data\_Buffer = 4 соответствует 27652

## Работа с модулем аналоговых входов MX04AD

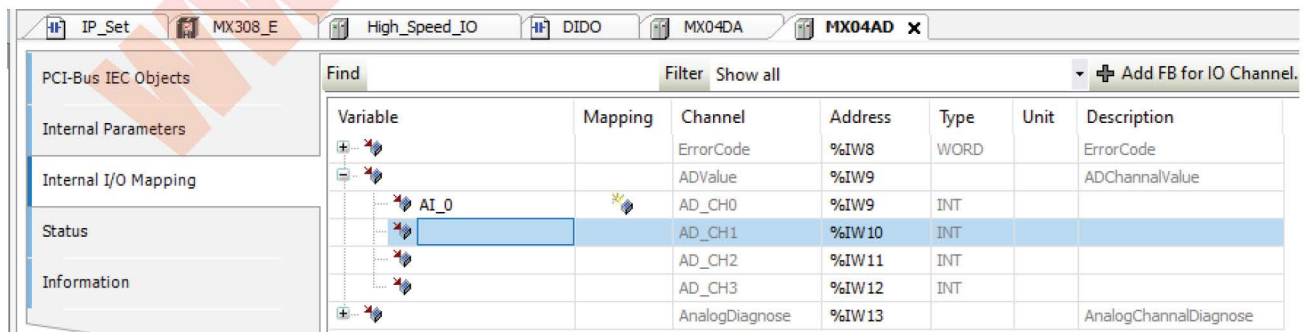
Добавьте в проект по вышеприведённой последовательности модуль MX04AD:



В проекте появится пункт:



Щёлкните на пункте MX04AD дважды левой кнопкой мышки, откроется вкладка параметров модуля. Выберите в ней пункт Internal I/O Mapping, в которой будут видны физические адреса выходов и можно задать тэги для выходов. Для этого разверните список и введите нужное название тега.



В программе к аналоговым входам можно обращаться как по тегам, так и по физическому адресу.

Для выбора режима работы аналогового входа необходимо открыть вкладку **Internal Parameters**. Выберите раздел типа **AD0 Config**. Для каждого канала AD0 – AD3 предусмотрен свой раздел. В данном разделе выберите пункт **Data Buffer**.

Parameter	Type	Value	Default Value
LocalBusSlave Info			
SlaveAddr	BYTE	0	0
SlaveNodeID	dword	1627389989	1627389989
AD0 config			
Index	UINT	16#8000	16#8000
SubIndex	UINT	16#01	16#01
Data_Buffer	UINT	0	0
Data_Size	UINT	1	1

В пункт **Data Buffer** вносится номер режима работы в соответствии с нижеприведённой таблицей:

Значение в Data_Buffer	Тип аналогового сигнала	Диапазон цифровой шкалы
0	-5~5V,	-32000~32000
1	1~5V,	0~32000
2	-10~+10V	-32000~32000
3	0~10V	0~32000
4	0~20mA	0~32000
5	4~20mA	0~32000
6	0~5V	0~32000
7	-20mA~20mA	-32000~32000

Далее, в разделе типа **AD0 filter config** в пункте **Data Buffer** нужно выставить цикл опроса входа в мс. По умолчанию стоит 4 мс.

AD0 filter config			
Index	UINT	16#8001	16#8001
SubIndex	UINT	16#01	16#01
Data_Buffer	UINT	4	4
Data_Size	UINT	1	1

Модуль можно конфигурировать и из программы. Для этого используются команды типа:

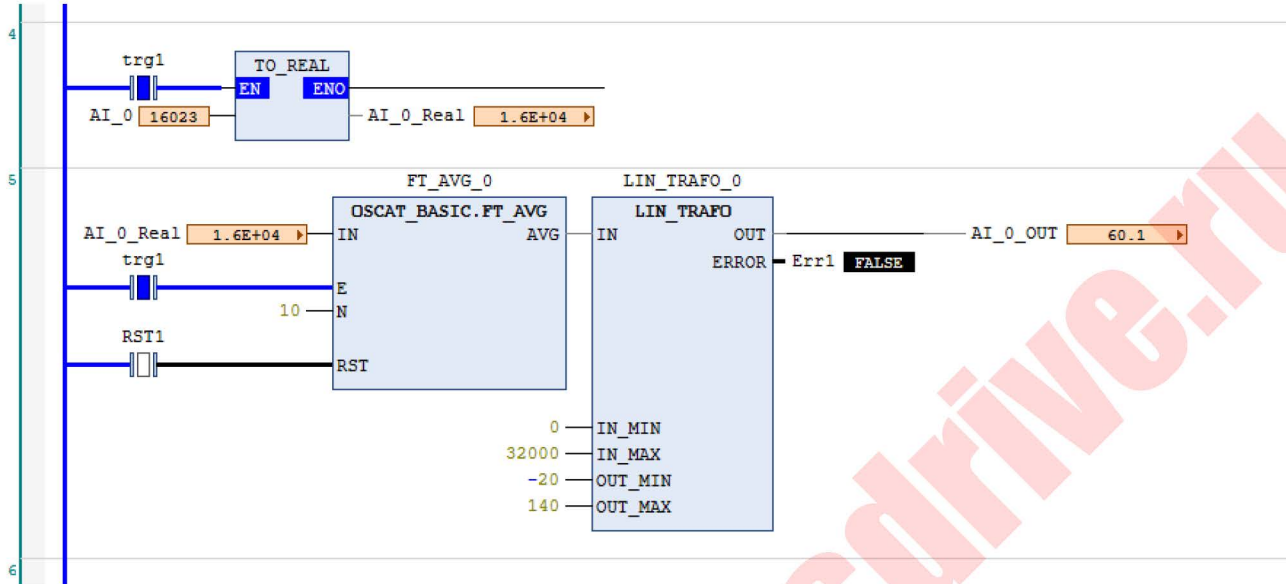
**LS\_ReadLocalBusSDO**

**LS\_WriteLocalBusSDO**

(см. пример в разделе модуля аналоговых выходов)



Для осреднения входных значений по принципу «бегущее среднее» можно использовать ФБ FT\_AVG из библиотеки OSCAT\_BASIC. Для масштабирования шкалы АЦП в шкалу требуемой физической величины можно использовать ФБ LIN\_TRAFO из библиотеки Util.



Состав библиотек можно посмотреть в Library Manager (дерево проекта).

Name	Namespace	Effective Version
3SLicense = 3SLicense, 3.5.18.0 (3S - Smart Software Solutions GmbH)	_3S_LICENSE	3.5.18.0
BreakpointLogging = Breakpoint Logging Functions, 3.5.17.0 (3S - Smart Software Solutions GmbH)	BPLog	3.5.17.0
CAA Device Diagnosis = CAA Device Diagnosis, 3.5.15.0 (CAA Technical Workgroup)	DED	3.5.15.0
<b>CAA NetBaseSrv = CAA Net Base Services, 3.5.15.0 (CAA Technical Workgroup)</b>	<b>NBS</b>	<b>3.5.15.0</b>
IoStandard = IoStandard, 3.5.15.0 (System)	IoStandard	3.5.15.0
LS_BasicModule = LS_BasicModule, 1.0.0.5 (Leadshine Technology Co.Ltd)	LS_BasicModule	1.0.0.5
MC_HSIO = MC_HSIO, 1.0.2.3 (Leadshine Technology Co.Ltd)	MC_HSIO	1.0.2.3
MC_SysLib = MC_SysLib, 1.0.0.8 (Leadshine Technology Co.Ltd)	MC_SysLib	1.0.0.8
<b>OSCAT_BASIC = BASIC, 3.3.4.0 (OSCAT)</b>	<b>OSCAT_BASIC</b>	<b>3.3.4.0</b>
SM3_Basic = SM3_Basic, 4.10.0.0 (3S - Smart Software Solutions GmbH)	SM3_Basic	4.10.0.0
SM3_CNC = SM3_CNC, 4.10.0.0 (3S - Smart Software Solutions GmbH)	SM3_CNC	4.10.0.0
SM3_Robotics = SM3_Robotics, 4.10.0.0 (3S - Smart Software Solutions GmbH)	SM3_Robotics	4.10.0.0
SM3_Robotics_Visu = SM3_Robotics_Visu, 4.10.0.0 (3S - Smart Software Solutions GmbH)	SM3_Robotics_Visu	4.10.0.0
SM3_Transformation = SM3_Transformation, 4.10.0.0 (3S - Smart Software Solutions GmbH)	TRAFO	4.10.0.0
<b>Util = Util, 3.5.18.0 (System)</b>	<b>Util</b>	<b>3.5.18.0</b>

При отсутствии данных библиотек их необходимо установить и подключить к проекту.

## Поддерживаемые базовые типы данных

Категории данных	Тип данных	Ключевые слова	Количество бит занимаемой памяти	Диапазоны
Логический тип	логическое значение	BIT	1	0 или 1
	логическое значение	BOOL	8	ЛОЖЬ(0) или ИСТИНА(1)
Целочисленный тип	байт	BYTE	8	0 ~ 255
	слово	WORD	16	0 ~ 65535
	двойное слово	DWORD	32	0 ~ 4294967295
	длинное слово	LWORD	64	0 ~ (2 <sup>64</sup> -1)
	короткий	SINT	8	- 128 ~ 127
	короткий беззнаковый	USINT	8	0 ~ 255
	целое число	INT	16	- 32768 ~ 32767
	целое число без знака	UINT	16	0 ~ 65535
	двойное целое число	DINT	32	- 2147483648 ~ 2147483647
	беззнаковое двойное целое	UDINT	32	0 ~ 4294967295
	длинное целое	LINT	64	-2 <sup>63</sup> ~ (2 <sup>63</sup> - 1)
	длинное целое число без знака	ULINT	64	0 ~ (2 <sup>64</sup> -1)
Тип с плавающей точкой	одинарная точность	REAL	32	1.175494351e- 38 ~ 3.402823466e+38
	двойная точность	LREAL	64	2.2250738585072014e- 308 ~ 1.7976931348623158e+308
Строка	строка	STRING	8*N бит	
Строка	строка Юникод	WSTRING	16*N бит	
Время		TIME	32	T#0ms~T#71582m47s295ms
Время дня		TIME_OF_DAY	32	TOD#0:0:0~TOD#1193:02:47.295
Дата		DATE	32	D#1970-1-1~D#2106-02-06
Дата и Время		DATE_AND_TIME	32	DT#1970-1-1-0:0:0~DT#2106-02-06-06: 28:15

### Примечание.

DesignerAX и CODESYS 3.5.18.30 и выше версиями поддерживают также новые типы данных и интерфейсов. В новых версиях среды программирования ряд инструкций поддерживают только новые типы данных

## Список наиболее употребительных команд

Среда программирования CODESYS является развитым, но достаточно сложным продуктом. В состав данного программного входит большое количество различных библиотек, каждая из которых содержит определённый набор команд. Существует несколько сообществ, разрабатывающих библиотеки независимо друг от друга. Кроме того, каждый производитель контроллеров разрабатывает и свой набор команд, работающих только с контроллерами данного производителя.

Контроллеры серии MX300 могут работать с библиотеками основного разработчика компании 3S, сообществ САА и OSCAT, а также с специализированной библиотекой производителя контроллеров. Команды из библиотеки производителя контроллеров начинаются с префикса LS\_ и устанавливаются в составе пэкиджа.

Для использования конкретной команды необходимо в состав проекта включить соответствующую библиотеку. Найти название библиотеки по имени команды можно в справке среды программирования (Help) или на сайте:

<https://www.helpme-codesys.com/>

Например, Вы хотите использовать команду BLINK (фликер). Изначально данная команда не находится при попытке её использовать, так как библиотека, в состав которой она входит, не подключена к проекту. Вы заходите на сайт <https://www.helpme-codesys.com/>, открывается страничка, на которой есть поле поиска, куда необходимо ввести название команды и нажать Enter.

## CODESYS Online Help

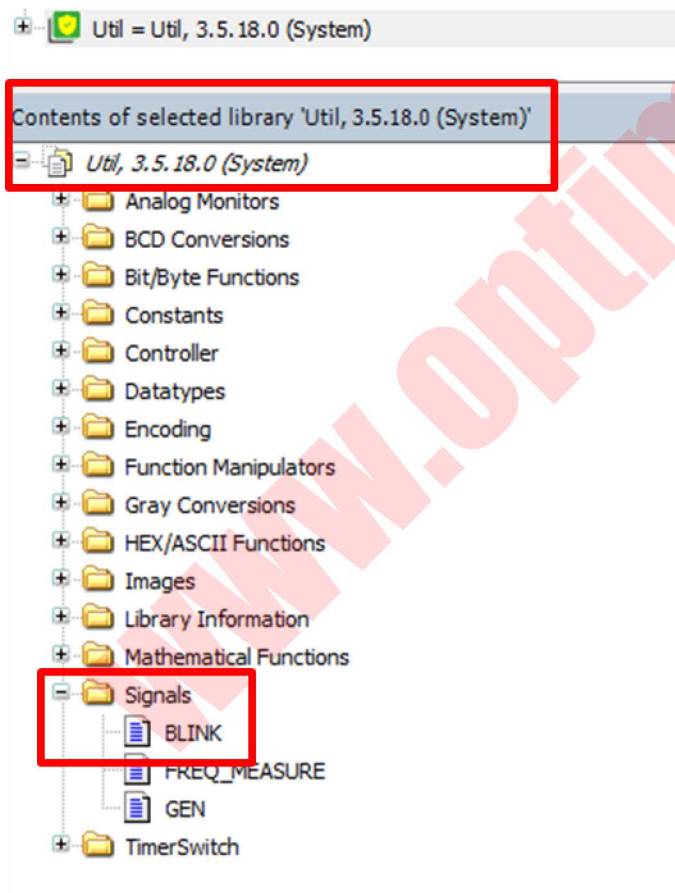
Welcome to the official CODESYS help site. Here, CODESYS Group experts have compiled answers to the most important questions from all CODESYS areas.

You have two options: Click through the topic tiles, or enter your search term directly into the search window.

В ответ будет показано описание команды и сверху будет видно название библиотеки:



Далее через Мастер установки библиотек подключите данную библиотеку (Signals в нашем примере) к своему проекту. Часто библиотека может быть в составе более общей библиотеки. В нашем примере библиотеке **Signals** входит в состав библиотеки **Utils**. В итоге библиотека и команда должны появиться в списке подключенных библиотек:



Для удобства работы с контроллерами серии MX300 далее приводится список наиболее употребительных команд.

Категория инструкции	Название инструкции	FB/FC	Описание
Сравнение	GT	FC	больше, чем
	LT	FC	меньше, чем
	GE	FC	больше или равно
	LE	FC	меньше или равно
	EQ	FC	равно
	NE	FC	не равно
Выбор	SEL	FC	выбор из 2-х значений по логическому состоянию
	MUX	FC	выбор по номеру из набора значений
	MAX	FC	выбор большего из 2-х чисел
	MIN	FC	выбор меньшего из 2-х чисел
	LIMIT	FC	Ограничение по верхнему и нижнему значению
Счётчики общего назначения	CTD	FB	Счёт вверх
	CTU	FB	Счёт вниз
	CTUD	FB	Счёт вверх-вниз
Таймеры	TP	FB	Таймер с импульсным запуском
	TON	FB	Таймер с задержкой на включение
	TOF	FB	Таймер с задержкой на выключение
	RTC	FB	Часы реального времени
Логические операции	AND	FC	побитовое И
	OR	FC	побитовое ИЛИ
	NOT	FC	Логическое НЕ
	XOR	FC	побитовое ЛИБО
	SR	FB	Бистабильное реле с приоритетом пуска
	RS	FB	Бистабильное реле с приоритетом стопа
	R_TRIG	FC	Обнаружение нарастающего фронта
	F_TRIG	FC	Обнаружение спадающего фронта
	SHR	FC	Побитовый сдвиг

Сдвиг данных			вправо
	SHL	FC	Побитовый сдвиг влево
	ROR	FC	Побитовое вращение вправо
	ROL	FC	Побитовое вращение влево

Категория инструкции	Название инструкции	FB/FC	Описание
Преобразование типов данных	BOOL_TO_<TYPE>	FC	BOOL в другой тип
	BYTE_TO_<TYPE>	FC	BYTE в другой тип
	WORD_TO_<TYPE>	FC	WORD в другой тип
	DWORD_TO_<TYPE>	FC	DWORD в другой тип
	INT_TO_<TYPE>	FC	INT в другой тип
	SINT_TO_<TYPE>	FC	SINT в другой тип
	DINT_TO_<TYPE>	FC	DINT в другой тип
	UDINT_TO_<TYPE>	FC	UDINT в другой тип
	REAL_TO_<TYPE>	FC	REAL в другой тип
	STRING_TO_<TYPE>	FC	STRING в другой тип
	TIME_TO_<TYPE>	FC	TIME в другой тип
	TOD_TO_<TYPE>	FC	TOD в другой тип
	DATE_TO_<TYPE>	FC	DATE в другой тип
	DT_TO_<TYPE>	FC	DT в другой тип
Инструкции конвертации данных	MOVE	FC	Присвоение значения
	HEXinASCII_TO_BYTE	FC	Шестнадцатеричное значение, записанное ASCII кодами, преобразует в двоичное число размером байт
	BYTE_TO_HEXinASCII	FC	Преобразует двоичное число размером байт в шестнадцатеричное число, записанное ASCII кодами
	WORD_AS_STRING	FC	ASCII коды в string
	BYTE_TO_HEXSTRING	FC	Байт в шестнадцатеричный стринг

	WORD_TO_HEXSTRING	FC	Слово в шестнадцатеричный стринг
	DWORD_TO_HEXSTRING	FC	Двойное слово в шестнадцатеричный стринг

Категория инструкций	Название инструкции	FB/FC	Описание
Математические инструкции	ADD	FC	Сложение
	SUB	FC	Вычитание
	MUL	FC	Умножение
	DIV	FC	Деление
	MOD	FC	Остаток от деления
	ABS	FC	Абсолютная величина
	SQRT	FC	Корень квадратный
	LN	FC	Натуральный логарифм
	LOG	FC	Десятичный логарифм
	EXP	FC	Возведение в степень числа e (e в степени x)
	EXPT	FC	Возведение в степень
	SIN	FC	Синус угла в радианах
	COS	FC	Косинус угла в радианах
	TAN	FC	Тангенс угла в радианах
	ASIN	FC	Арксинус угла в радианах
	ACOS	FC	Арккосинус угла в радианах
	ATAN	FC	Арктангенс угла в радианах
	XSIZEOF	FC	Определяет необходимое количество байтов для хранения данных

Категория инструкций	Название инструкции	FB/FC	Описание
Строковые инструкции	LEN	FC	Количество символов в строке
	LEFT	FC	Возвращает заданное количество символов при отсчёте слева
	RIGHT	FC	Возвращает заданное количество символов при отсчёте справа
	MID	FC	Возвращает заданное количество символов с заданной позиции в строке
	CONCAT	FC	Конкатенация (соединение) двух строк
	INSERT	FC	Вставка строки в заданную позицию другой строки
	DELETE	FC	Удаление заданного количества символов с заданной позиции в строке
	FIND	FC	Определение наличия одной строки в другой
	REPLACE	FC	Заменяет заданное количество символов с указанного места в строке

Категория инструкций	Название инструкции	FB/FC	Описание
Операции с файлами	SysFileOpen	FB	Открыть файл
	SysFileClose	FB	Закрыть файл
	SysFileWrite	FB	Записать файл
	SysFileRead	FB	Прочитать файл
	SysFileDelete	FB	Удалить файл
	SysFileCopy	FB	Копировать файл
	SysFileRename	FB	Переименование файла
	SysFileSetPos	FB	Установить местоположение чтения и записи файла
	SysFileGetPos	FB	Получить местоположение чтения и записи файла
	SysFileGetSize	FB	Получить размер файла



Категория инструкций	Название инструкции	FB/FC	Описание
Регуляторы	PD	FB	ПД регулятор
	PID	FB	ПИД регулятор
	PID_FIXCYCLE	FB	ПИД регулятор с установкой цикла опроса
Преобразование BCD	BCD_TO_INT	FC	Преобразование числа в формате BCD в тип INT
	INT_TO_BCD	FC	Преобразование числа типа INT в формат BCD
	BCD_TO_BYTE	FC	Преобразование числа в формате BCD в тип BYTE
	BYTE_TO_BCD	FC	Преобразование числа типа BYTE в формат BCD
	BCD_TO_WORD	FC	Преобразование числа в формате BCD в тип WORD
	WORD_TO_BCD	FC	Преобразование числа тип WORD в формат BCD
	BCD_TO_DWORD	FC	Преобразование числа в формате BCD в тип DWORD
	DWORD_TO_BCD	FC	Преобразование числа тип DWORD в формат BCD
Программные фликеры	BLINK	FB	Программный генератор импульсов
	GEN	FB	Программный генератор периодических сигналов
	FREQ_MEASURE	FB	Измерение частоты входных импульсов

Категория инструкций	Название инструкции	FB/FC	Описание
Состояние оси	MC_Power	FB	Включение оси
	MC_Reset	FB	Сброс оси
	MC_ReadStatus	FB	Чтение статуса оси
	MC_ReadAxisError	FB	Чтение кода ошибки оси
	MC_ReadParameter	FB	Чтение параметров
	MC_ReadBoolParameter	FB	Чтение логических параметров
	MC_WriteParameter	FB	Запись параметров
	MC_WriteBoolParameter	FB	Запись логических параметров
	MC_ReadActualPosition	FB	Чтение положения оси
	MC_ReadActualVelocity	FB	Чтение скорости оси
	MC_ReadActualTorque	FB	Чтение момента на оси
	MC_SetPosition	FB	Задание позиции

SMC_ReadSetPosition	FB	Чтение заданной позиции
SMC_ReadFBError	FB	Чтение исторической информации об ошибках
SMC_ClearFBError	FB	Очистить исторические сообщения об ошибках
SMC_ErrorString	FB	Текст ошибки

Категория инструкций	Название инструкции	FB/FC	Описание
Управление одноосевым движением	MC_Home	FB	Возврат в ноль
	MC_MoveAbsolute	FB	Абсолютное позиционирование
	MC_MoveRelative	FB	Относительное позиционирование
	MC_MoveVelocity	FB	Движение с фиксированной скоростью
	MC_Stop	FB	Стоп движения оси
	MC_Halt	FB	Пауза движения оси
	MC_Jog	FB	Джог-движение
	MC_MoveAdditive	FB	Аддитивное позиционирование
	MC_MoveSuperImpose	FB	Наложённое движение
	MC_PositionProfile	FB	Плановое движение
	MC_VelocityProfile	FB	Плановое движение в режиме скорости
	MC_AccelerationProfile	FB	Плановое ускорение
	SMC_Homing	FB	Возврат в нулевую позицию
	SMC_Inch	FB	Пошаговое движение оси
Управление групповым движением осей	MC_GearIn	FB	Электронный редуктор
	MC_GearInPos	FB	Электронный редуктор с учётом соотношения позиций осей
	MC_GearOut	FB	Расцепление осей (прекращение MC_GearIn)
	MC_CamTableSelect	FB	Выбор таблицы E-CAM
	MC_CamIn	FB	Запуск конкретного E-CAM
	MC_CamOut	FB	Отключение конкретного E-CAM
	SMC_GetTappetValue	FB	Отображение статуса текущего сегмента E-CAM
	LS_2AxisLine	FB	Двухосевая линейная интерполяция

Управление интерполирова нным движением осей	LS_3AxisLine	FB	Трёхосевая линейная интерполяция
	LS_4AxisLine	FB	Четырёхосевая линейная интерполяция
	LS_5AxisLine	FB	Пятиосевая линейная интерполяция
	LS_6AxisLine	FB	Шестиосевая линейная интерполяция
	LS_2AxisLineA_Ratio	FB	Двухосевая линейная интерполяция с регулируемой скоростью
	LS_LineFollow	FB	Отслеживание
	LS_2AxisCircle	FB	Двухосевая круговая интерполяция
	LS_3AxisCircle	FB	Трёхосевая круговая интерполяция
	LS_2AxisEllipses	FB	Двухосевая эллиптическая интерполяция
	LS_2AxisCircle_Helical	FB	Спиральная интерполяция
	LS_3AxisMoveSequence	FB	Трёхосное непрерывное интерполяционное движение
	LS_4AxisMoveSequence	FB	Четырёхосное непрерывное интерполяционное движение
	LS_6AxisMoveSequence	FB	Шестиосное непрерывное интерполяционное движение

Категория инструкций	Название инструкции	FB/FC	Описание
TCP сокет	TCP_Client	FB	Открыть TCP сокет в режиме клиента
	TCP_Write	FB	Отправка данных через TCP сокет
	TCP_Read	FB	Приём данных через TCP сокет
	TCP_Connection	FB	Инкапсуляция TCP соединения между клиентом и сервером
	TCP_Server	FB	Открыть TCP сокет в режиме сервера
UDP сокет	UDP_Peer	FB	Открыть UDP сокет
	UDP_Receive	FB	Приём данных через UDP сокет
	UDP_Send	FB	Отправка данных через UDP сокет
	ETC_CO_SdoReadDWord	FB	Чтение данных EtherCAT через SDO, результат в формате DWORD

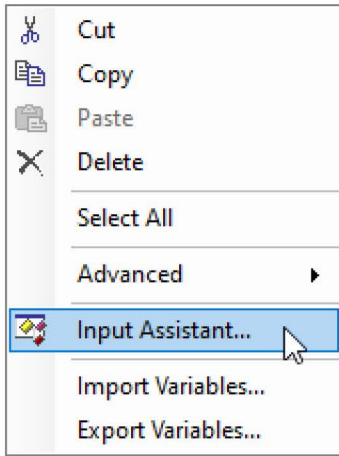
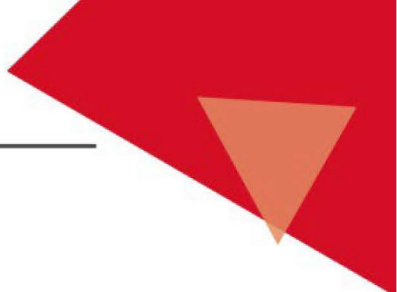
Инструкции EtherCAT	ETC_CO_SdoRead4	FB	Чтение данных EtherCAT через SDO, результат в формате ARRAY [1..4] OF BYTE
	ETC_CO_SdoRead	FB	Чтение данных EtherCAT через SDO, результат в формате CAA.SIZE
	ETC_CO_SdoWrite_DWord	FB	Запись данных EtherCAT через SDO, данные в формате DWORD
	ETC_CO_SdoWrite4	FB	Запись данных EtherCAT через SDO, данные в формате ARRAY [1..4] OF BYTE
	ETC_CO_SdoWrite	FB	Запись данных EtherCAT через SDO, данные в формате CAA.SIZE
	IoDrvEthercat_Diag	FB	Расширение для IoDrvEtherCAT, предоставляющее диагностические функции
	ETCSlave	FB	Создание экземпляра типа ETCSlave для каждого ведомого устройства EtherCAT в древе проекта
Инструкции Ethernet/IP	IoDrvEtherNetIP_Diag	FB	Расширение для IoDrvEtherNetIP
	RemoteAdapter	FB	Расширения для LAT.Element
	Generic_Service	FB	Предоставляет generic service для EtherNet/IP Adapter
	Get_Attributes_All	FB	Получить все свойства экземпляра объекта
	Get_Attribute_Single	FB	Получить один атрибут экземпляра объекта
	Set_Attributes_All	FB	Установить все свойства экземпляра объекта
	Set_Attribute_Single	FB	Установите одно свойство экземпляра объекта

Категория инструкций	Название инструкции	FB/FC	Описание
Импульсное управление осью	LS_Home_P	FB	Импульсная ось возвращается в исходное положение
	LS_MotionControl_P	FB	привязка оси импульса
	LS_ReadAxisPara_P	FB	Получите эквивалентное значение импульса оси импульса

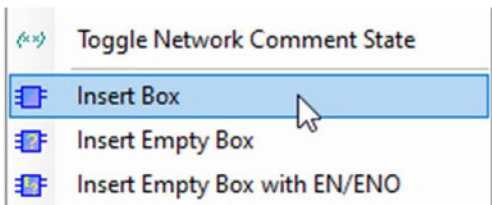
	LS_ResetAxis_P	FB	Сброс оси импульса
Высокоскоростной счетчик	LS_Counter	FB	Инструкция высокоскоростного счетчика
	LS_PresetValue	FB	Команда предустановленного значения счетчика

Категория инструкций	Название инструкции	FB/FC	Описание
Операции с SD картой	LS_CopyFromSDCard	FB	Копирование файлов с SD карты на локальный диск
	LS_CopyToSDCard	FB	Копирование локальных файлов на SD карту
	LS_GetSDCardInformation	FB	Получение информации о файле на SD карте
Системные команды	LS_GetIpAddress	FB	Прочитать IP адрес контроллера
	LS_SetIpAddress	FB	Установить IP адрес контроллера
	GetPLCLoad	FC	Прочитать степень загрузки контроллера
	GetPLCVersion	FC	Прочитать версию прошивки контроллера
	ColdResetApp	FC	«Холодная» перезагрузка контроллера
	WarmResetApp	FC	«Тёплая» перезагрузка контроллера
	LS_ReadDintDT	FB	Получить системное время в формате DINT
	LS_ReadStringDT	FB	Получить системное время в формате String
	LS_SetDintDT	FB	Установить системное время в формате DINT
LS_SetStringDT	FB	Установить системное время в формате String	

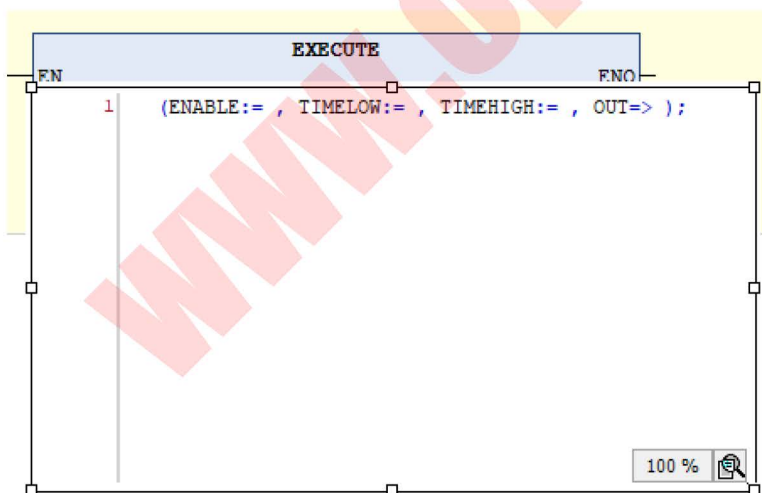
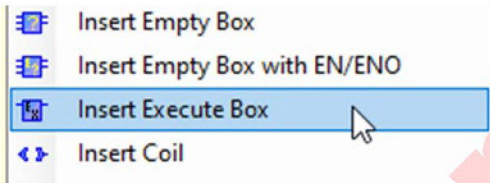
Для добавления в проект команды типа FB или FC на языке ST необходимо просто вызвать нажатием правой кнопки мышки и выбрать пункт **Input Assistant**:



На языке LD команды типа FB вводятся через меню



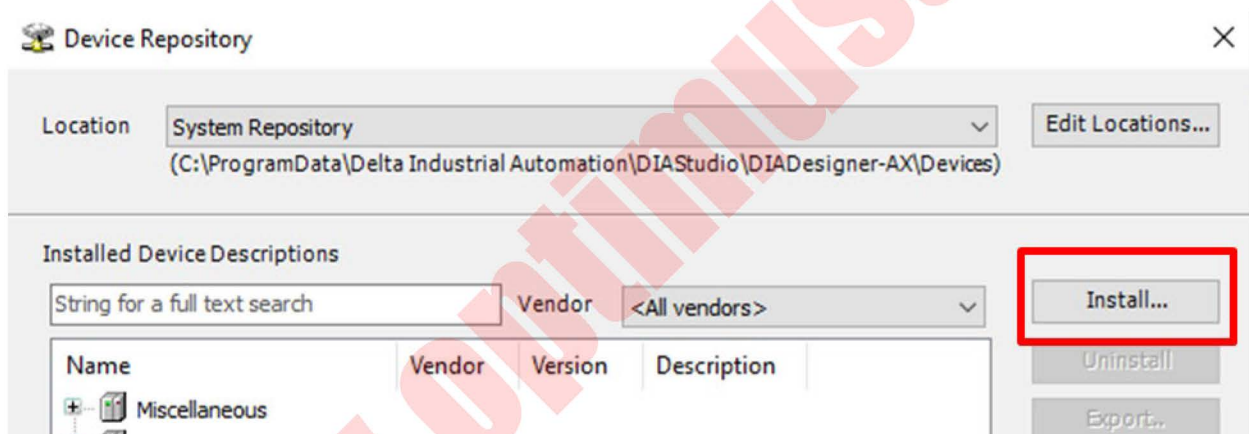
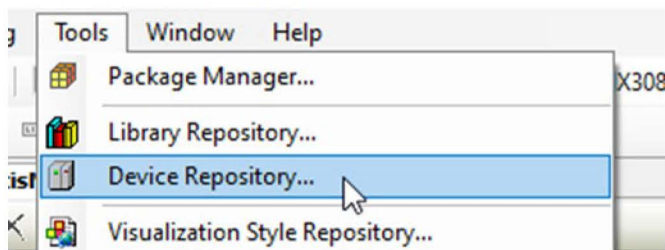
а команды типа FC вводятся через



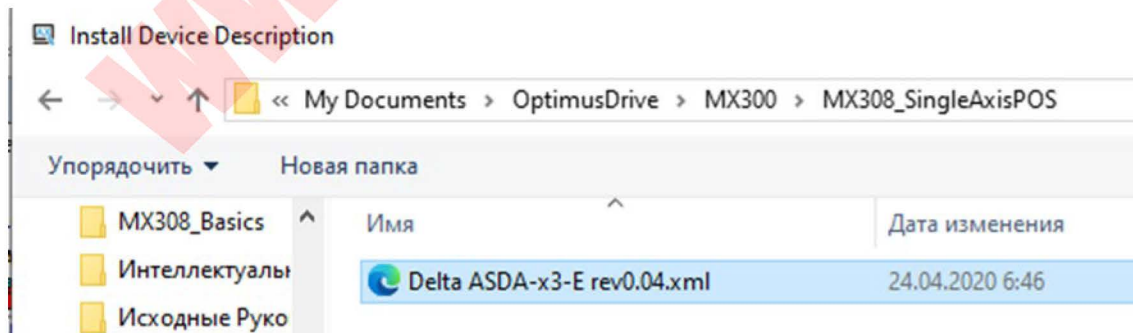
## Добавление в проект сервопривода

Контроллеры серии MX300 позволяют добавить в проект в качестве оси движения любой сервопривод с интерфейсом EtherCAT CoE (CANopen over EtherCAT) и поддерживающим стандарт CiA402, для которого производитель предоставляет корректный XML файл с описанием устройства. Сервопривод добавляется в проект как полноценная ось (Motion Axis), поддерживающая интерполированное движение. Вариант «лёгкой оси» для перемещений типа точка-точка не поддерживается.

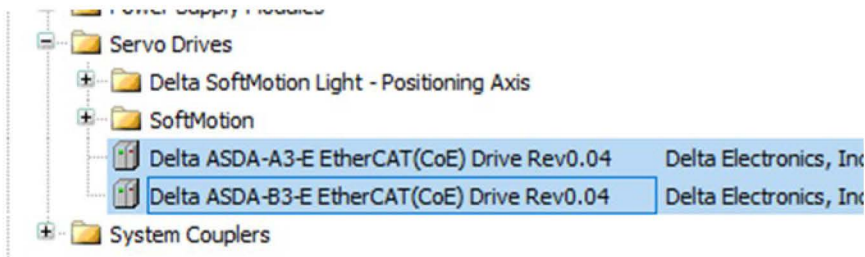
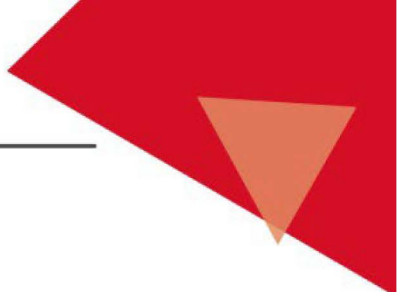
Для добавления в проект сервопривода сначала необходимо установить файл с описанием устройства в депозитарий (**Tools – Device Repository**):



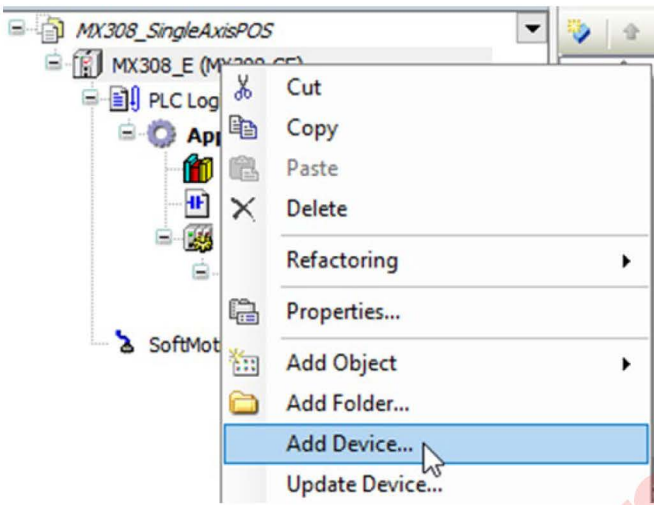
В качестве примера установим XML файл для сервопривода Delta ASD-B3-E:



После установки файла в списке устройств появятся две записи:



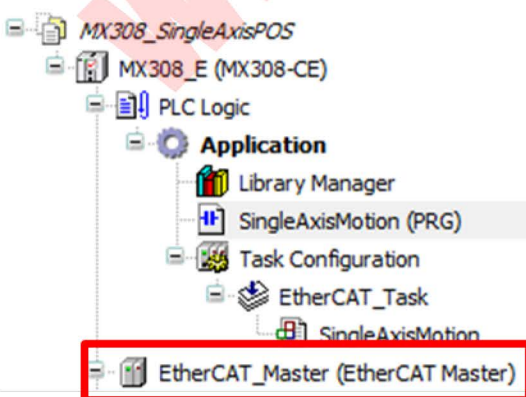
Далее необходимо добавить в проект EtherCAT адаптер. Для этого щёлкните правой кнопкой на названии контроллера и в меню выберите пункт **Add Device**:



В открывшемся окне выберите пункт **EtherCAT - Master - EtherCAT Master**:

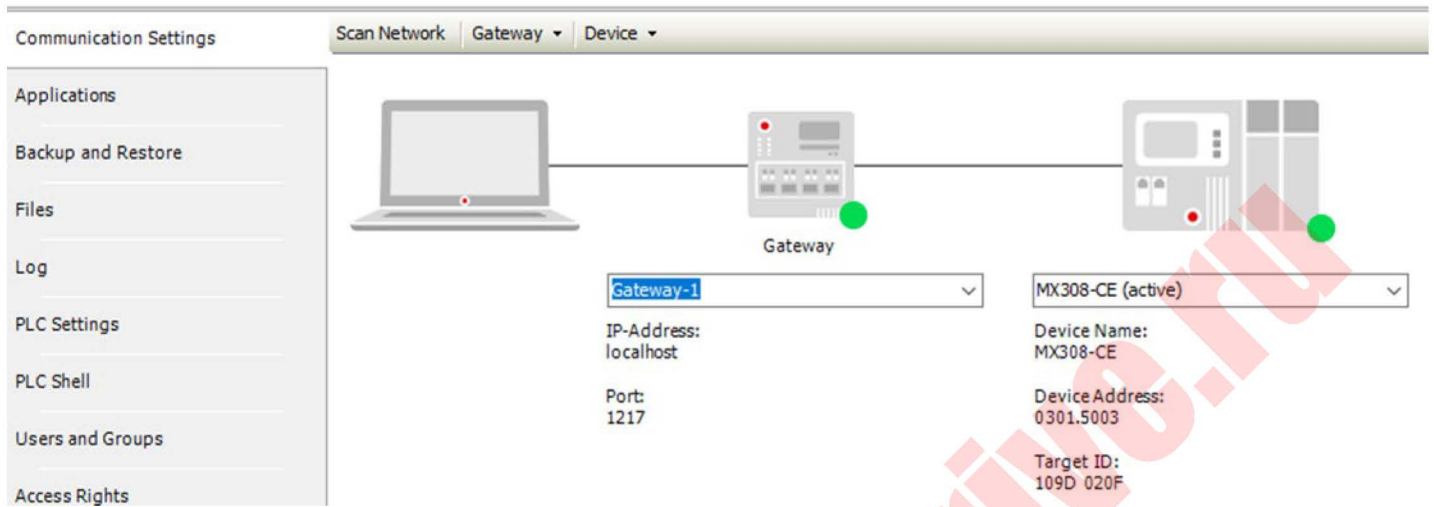


В древе проекта появится пункт:

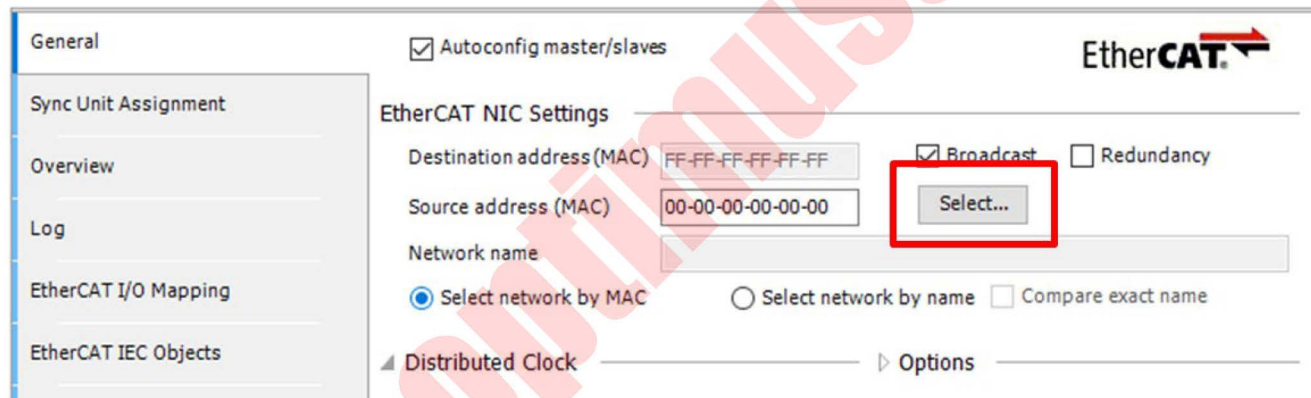




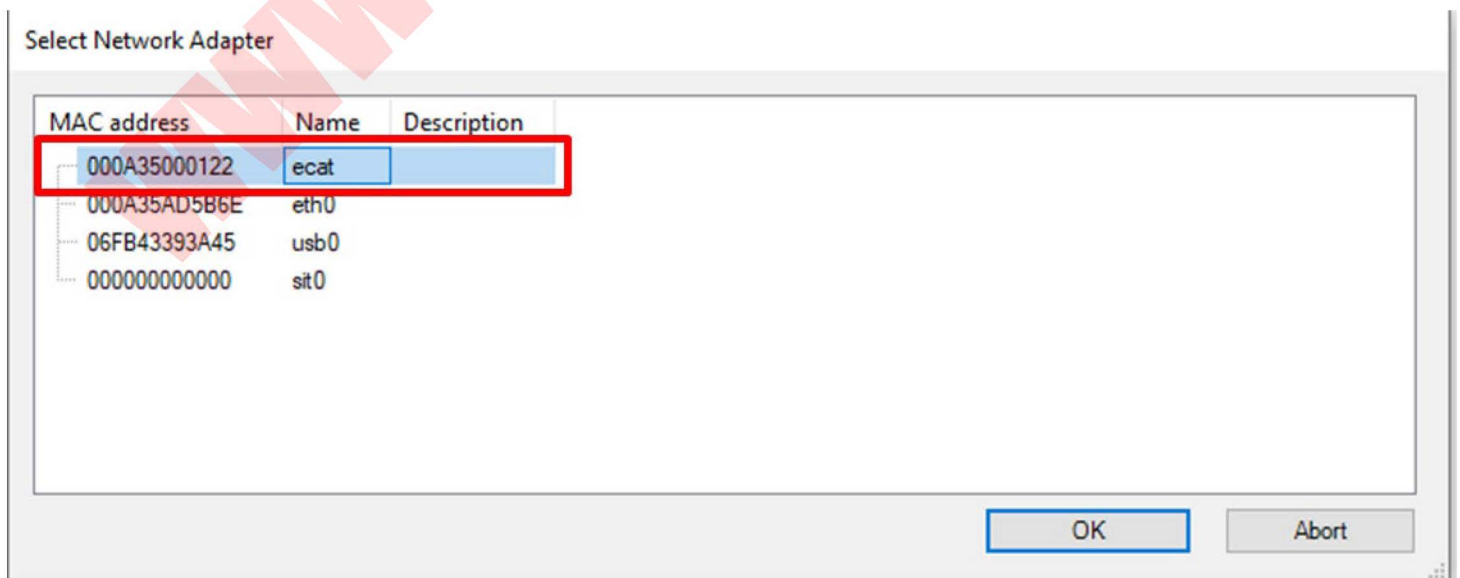
Далее необходимо назначить в проекте Мастера сети EtherCAT. В нашем примере это будет контроллер MX300, поэтому необходимо установить с ним связь (см. соответствующий раздел данного Руководства):



Далее щёлкните дважды левой кнопкой мышки на пункте **EtherCAT Master** и в открывшейся вкладке выберите пункт **General** и нажмите кнопку **Select**:



В открывшемся окне выберите пункт (MAC адрес у каждого контроллера свой):



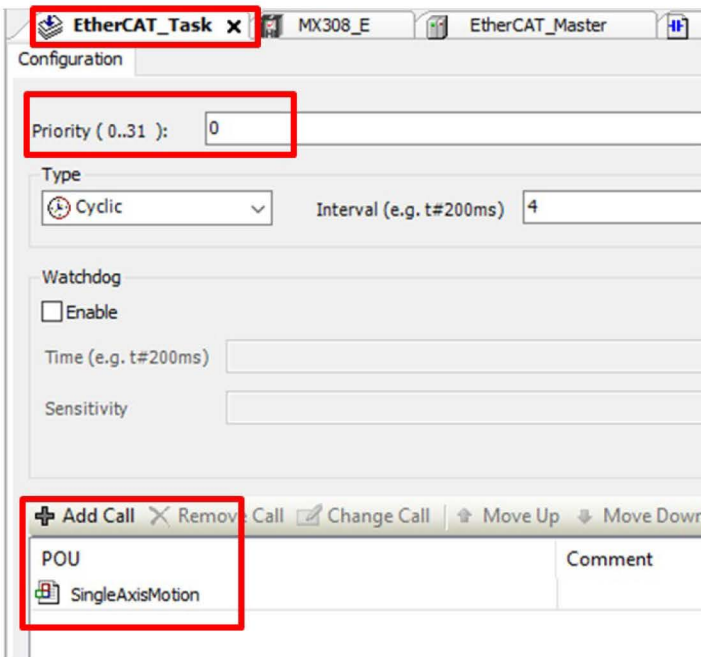
MAC адрес должен установиться в данном поле:

Далее необходимо активировать задачу EtherCAT\_Task. Для этого сделайте следующие настройки:

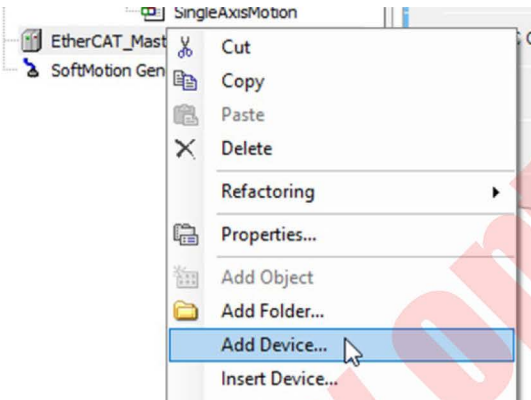
Во вкладке Контроллера:

Во вкладке EtherCAT\_Master:

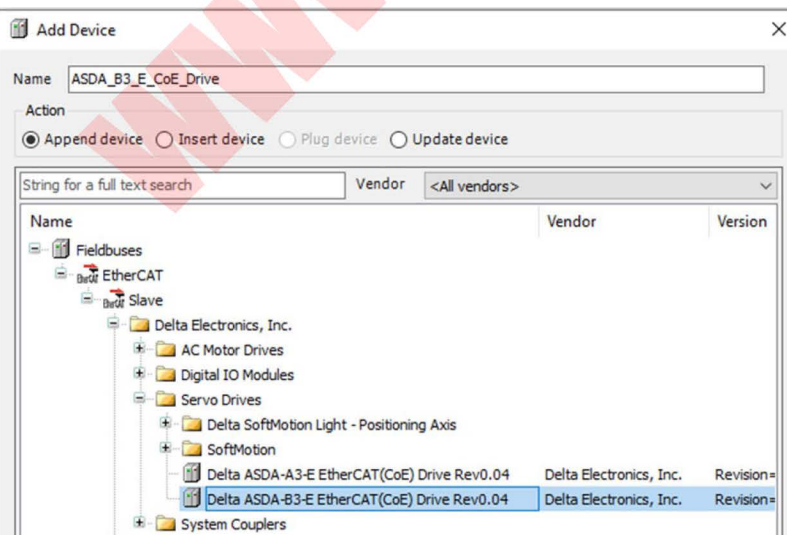
Во вкладке задачи выставить высший приоритет (0), привязать нужные POU:



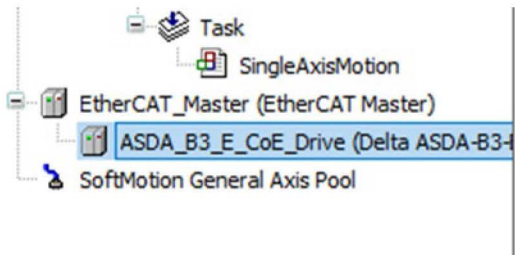
Далее необходимо добавить в проект сервопривод. Для этого щёлкните правой кнопкой мышки на пункте **EtherCAT Master** и в открывшемся окне выберите пункт **Add Device**:



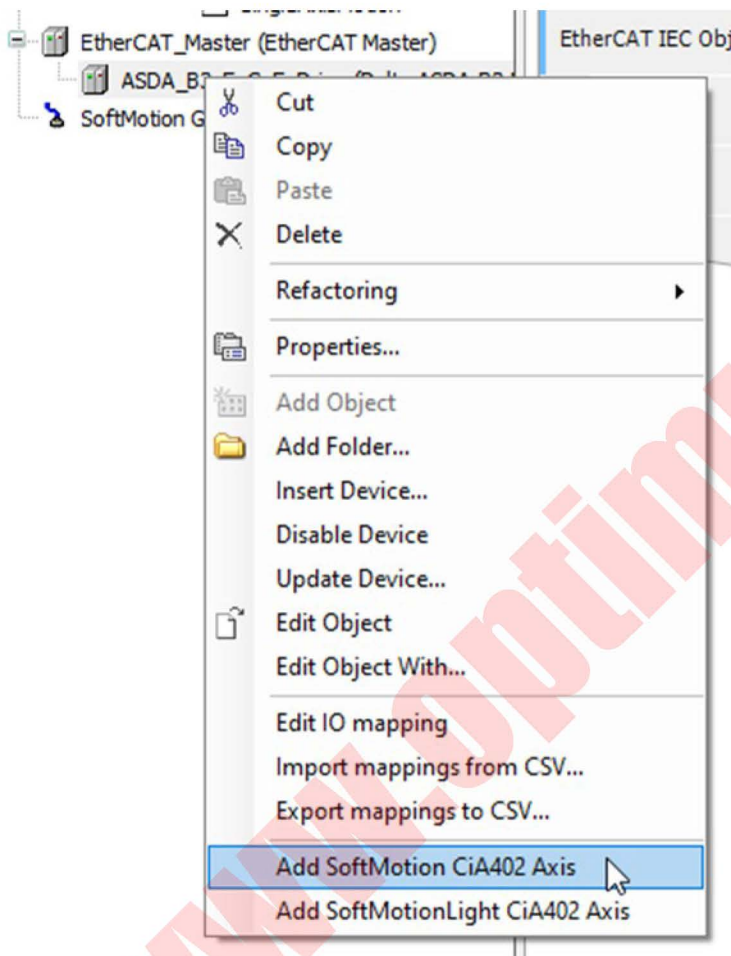
В открывшемся окне выберите сервопривод Delta ASD-B3-E:



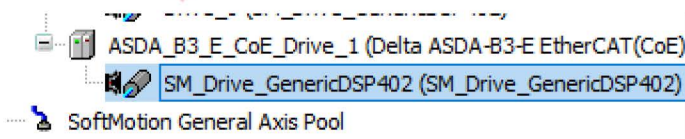
В древе проекта появится пункт:



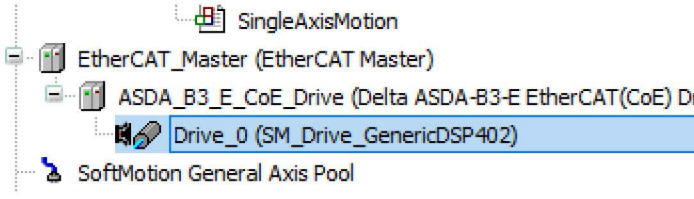
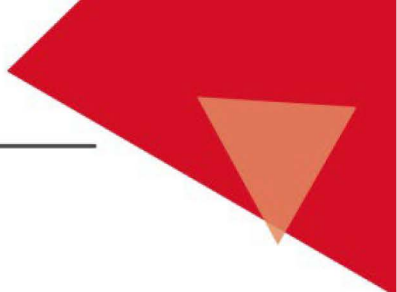
Далее к данному узлу необходимо добавить ось движения стандарта CiA402. Для этого щёлкните правой кнопкой мышки на пункте **ASDA\_B3\_E\_CoE\_Drive** и выберите пункт **Add SoftMotion CiA402 Axis**:



Появится пункт:



Название оси можно переименовать, например в Drive\_0:



Далее необходимо произвести настройку узла и оси движения.

Для настройки узла щёлкните на пункте **ASDA\_B3\_E\_CoE\_Drive** два раза левой кнопкой мышки и в отрывшейся вкладке выберите пункт **Process Data**, в котором отметьте нужные наборы PDO:

Name	Type	Index
<input type="checkbox"/> 16#1600 1st RxPDO Mapping (exclu		
Control Word	UINT	16#6040:16#00
Target Position	DINT	16#607A:16#00
Target Velocity	DINT	16#60FF:16#00
Touch Probe Function	UINT	16#60B8:16#00
<input checked="" type="checkbox"/> 16#1601 2nd RxPDO Mapping		
Control Word	UINT	16#6040:16#00
Target Position	DINT	16#607A:16#00
Target Velocity	DINT	16#60FF:16#00
Target Torque	INT	16#6071:16#00
Touch Probe Function	UINT	16#60B8:16#00
<input type="checkbox"/> 16#1602 3rd RxPDO Mapping (exclu		
Control Word	UINT	16#6040:16#00
Target Position	DINT	16#607A:16#00
Target Velocity	DINT	16#60FF:16#00
Target Torque	INT	16#6071:16#00
Mode Of Operation	SINT	16#6060:16#00

Name	Type	Index
<input type="checkbox"/> 16#1A00 1st TxPDO Mapping (e		
Status Word	UINT	16#6041:16#00
Actual Position	DINT	16#6064:16#00
Velocity actual value	DINT	16#606C:16#00
Touch Probe Status	UINT	16#60B9:16#00
Touch Probe Pos1 Pos.Value	DINT	16#60BA:16#00
DigitalInputs	UDINT	16#60FD:16#00
<input checked="" type="checkbox"/> 16#1A01 2nd TxPDO Mapping		
Status Word	UINT	16#6041:16#00
Actual Position	DINT	16#6064:16#00
Velocity actual value	DINT	16#606C:16#00
Actual Torque	INT	16#6077:16#00
Touch Probe Status	UINT	16#60B9:16#00
Touch Probe Pos1 Pos.Value	DINT	16#60BA:16#00
DigitalInputs	UDINT	16#60FD:16#00
<input type="checkbox"/> 16#1A02 3rd TxPDO Mapping (e		
Status Word	UINT	16#6041:16#00

Для настройки оси движения щёлкните на пункте **Drive\_0** два раза левой кнопкой мышки и в отрывшейся вкладке выберите пункт **General**.

**Axis type and limits**

Virtual mode  
 Modulo  
 Finite

**Software limits**

Activated

Negative [u]:   
 Positive [u]:

**Software error reaction**

Deceleration [u/s²]:   
 Max. distance [u]:

**Dynamic limits**

Velocity [u/s]:    
 Acceleration [u/s²]:    
 Deceleration [u/s²]:    
 Jerk [u/s³]:

**Velocity ramp type**

Trapezoid  
 Sin²  
 Quadratic  
 Quadratic (smooth)

**Identification**

ID:

**Position lag supervision**

deactivated (dropdown)  
 Lag limit [u]:

Так как в инструкциях управления движением все параметры, связанные с движением, задаются в единицах пользователя, то перед началом настройки оси необходимо вычислить все необходимые параметры движения и внести в разделы **General** и **Scaling/Mapping**.

Вначале необходимо заполнить ключевые параметры в разделе **Scaling/Mapping**:

General	Motor Type	Scaling
Scaling/Mapping	<input checked="" type="radio"/> Rotary	<input type="checkbox"/> Invert direction
Commissioning	<input type="radio"/> Linear	16#10000 increments <=> motor turns 1
SM_Drive_ETC_GenericDSP402: Parameters		1 motor turns <=> gear output turns 1
		1 gear output turns <=> units in application 1

В верхней строке в левом поле необходимо ввести количество импульсов на оборот сервопривода. Это должно строго совпадать с настройками привода.

Scaling

Invert direction

**16#10000** increments

В нашем примере используется сервопривод Delta ASD-B3-E, мотор которого имеет энкодер 24 бит. Это составляет 16 777 216 импульсов на оборот. По умолчанию числитель и знаменатель стоят 1. Значит в данное поле при таких настройках нужно ввести число 16777216 (на 1 оборот, т.е. поле справа остаётся равным 1).

Scaling

Invert direction

**16777216** increments <=> motor turns 1

Если к примеру установить в приводе вот такие числитель и знаменатель:

Числитель P1-044 = 16 777 216  
 Знаменатель P1-045 = 100 000

то мы получим число 100 000 импульсов на оборот, которое нужно будет ввести в поле:

Scaling

Invert direction

**100000** increments <=> motor turns 1

Следующее поле – это редуктор. Например, если у Вас понижающий редуктор 1:15, то нужно ввести такие числа:

**15** motor turns <=> gear output turns **1**

Во многих случаях редуктор не вводят, а учитывает его в конечном перемещении, которое вводится в третьей строчке. В нашем примере это 10 единиц.

1	gear output turns <=> units in application	10
---	--	----

Это ключевой параметр, который увязывает обороты мотора сервопривода и перемещение конечного механизма. В примере выше – 1 оборот мотора сервопривода вызывает перемещение конечного механизма (например ШВП) на 10 пользовательских единиц, которые могут соответствовать любым единицам длины (мкм, мм, дюймы и т.д.).

Таким образом, Вы устанавливаете жёсткую связь между заданием в единицах длины и единицами мотора (импульсами на оборот). В командах движения задание перемещения устанавливается в единицах пользователя, т.е. длины, и это автоматически пересчитывается в задание для мотора в его единицах (импульсы на оборот по внутреннему энкодеру).

В итоге для данного случая получатся следующие настройки:

General	Motor Type	Scaling		
Scaling/Mapping	<input checked="" type="radio"/> Rotary	<input type="checkbox"/> Invert direction		
Commissioning	<input type="radio"/> Linear	100000	increments <=> motor turns	1
SM_Drive_ETC_GenericDSP402: I/O Mapping		15	motor turns <=> gear output turns	1
		1	gear output turns <=> units in application	10

Для улучшения точности, перемещение конечного механизма на 1 оборот лучше измерять прямым способом высокоточным измерительным инструментом (микрометром) и уже на основе объективных данных устанавливать линейные единицы пользователя.

Также, при прямом измерении можно учесть редуктор (коэффициент редукции), т.е. учесть его в перемещении конечного механизма.

Кроме того, для устранения нарастающей погрешности при вычислениях перемещение конечного механизма лучше задавать в крупных числах. Например, вместо 10 мм использовать 10 000 мкм.

Для дальнейших вычислений скорости и ускорения в единицах пользователя примем следующие параметры линейного перемещения и количества импульсов на оборот двигателя:

General	Motor Type	Scaling		
Scaling/Mapping	<input checked="" type="radio"/> Rotary	<input type="checkbox"/> Invert direction		
Commissioning	<input type="radio"/> Linear	16777216	increments <=> motor turns	1
SM_Drive_ETC_GenericDSP402: I/O Mapping		1	motor turns <=> gear output turns	1
		1	gear output turns <=> units in application	10000

Расчёт максимальной скорости:

Максимальная скорость вращения мотора 3000 об/мин. В секунду  $3000 : 60 = 50$  об/сек

За один оборот будет пройдено 10 000 ед. длины в единицах пользователя

Следовательно максимальная скорость:  $50 * 10000 = 500\,000$  ед. в секунду

Это максимально допустимая скорость, которую можно задавать в инструкциях движения.

Расчёт максимально допустимого ускорения/замедления.

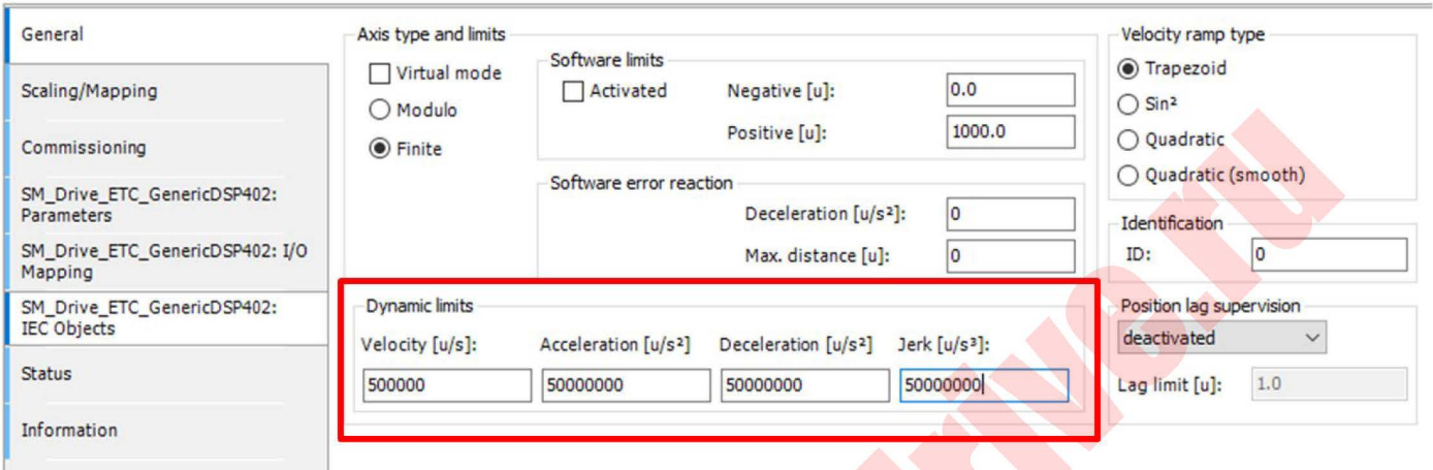
Сначала рассчитывается теоретически достижимое ускорение при разгоне до 3000 об/мин за 1 мс.

Исходя из полученных выше расчётов за 1 секунду мы можем достичь 500 000 ед. длины/сек. Для достижения такой скорости за 1 мс это значение надо умножить на 1000 (т.е. перевести секунды в мс).

Получается число 500 000 000 ед./сек<sup>2</sup>

Это предельно теоретически возможное ускорение в единицах пользователя, соответствующее разгону привода до 3000 об/мин за 1 мс.

Теперь необходимо выставить предельные значения скорости и ускорения во вкладке **General**:



В целях безопасности предельное ускорение уменьшено в 10 раз. Т.е. приводу разрешается разогнаться до 3000 об/мин как минимум за 10 мс (50 000 000 ед./сек<sup>2</sup>). Такое же значение нужно выставить и для параметра Jerk

В инструкциях движения позиция, скорость и ускорение задаются в единицах пользователя согласно полученных вычислений.

Например (при перемещении на 10000 ед. за 1 оборот двигателя):

Ускорение 100 мс = 5 000 000 ед/сек<sup>2</sup>

Ускорение 20 мс = 25 000 000 ед/сек<sup>2</sup>

Скорость 3000 об/мин = 500 000 ед/сек

Скорость 300 об/мин = 50 000 ед/сек

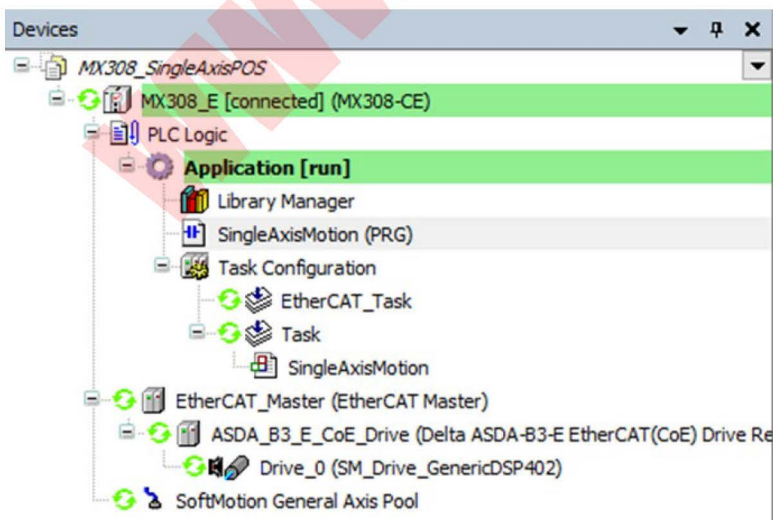
Скорость 15 об/мин = 2500 ед/сек

Перемещение на 10 мм = 10000 ед.

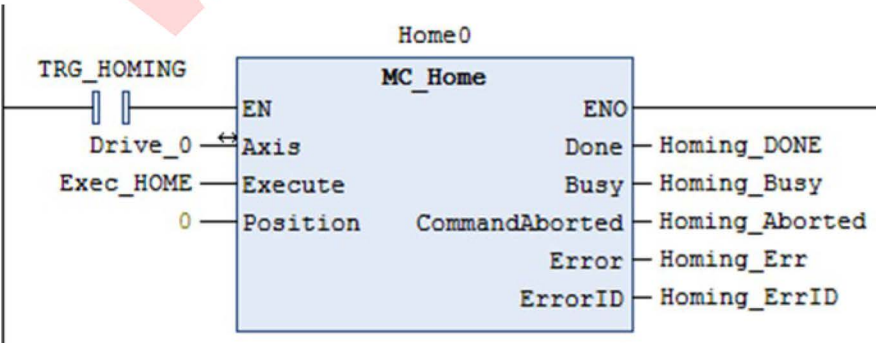
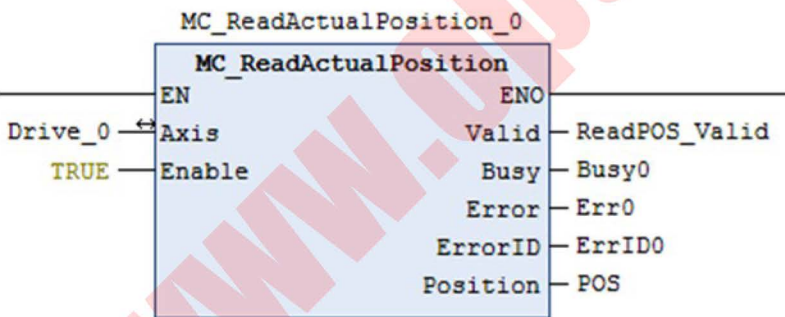
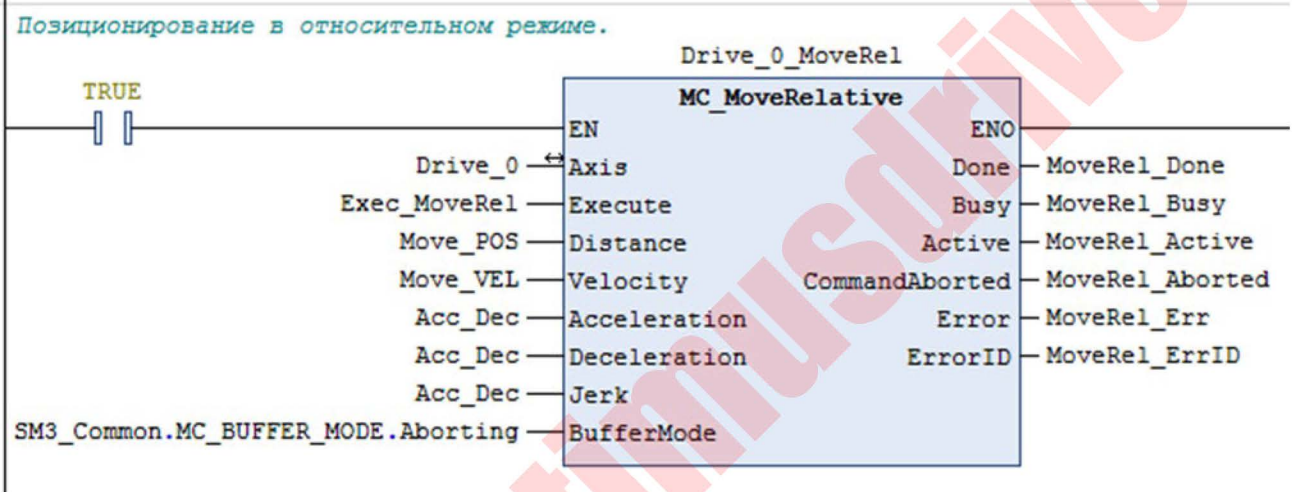
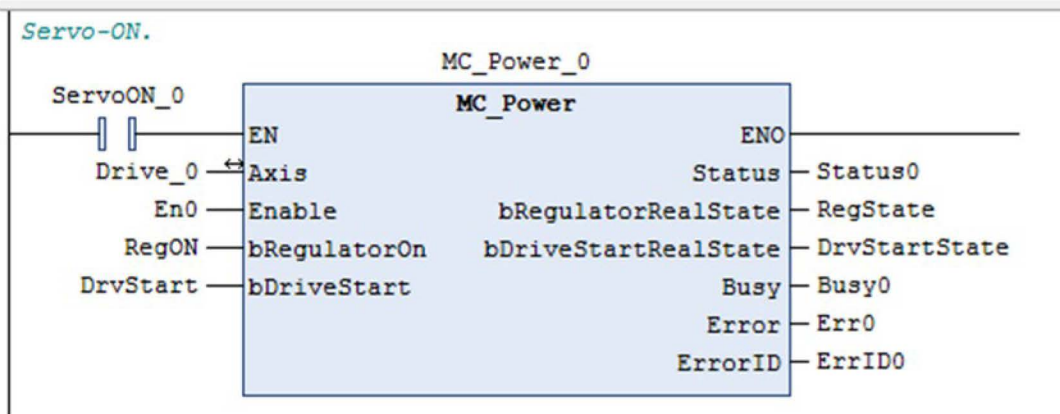
Перемещение на 20,456 мм = 20456 ед.

Перемещение на 300 мм = 300 000 ед.

Ниже приводятся для примера ряд инструкций для одноосевого движения:



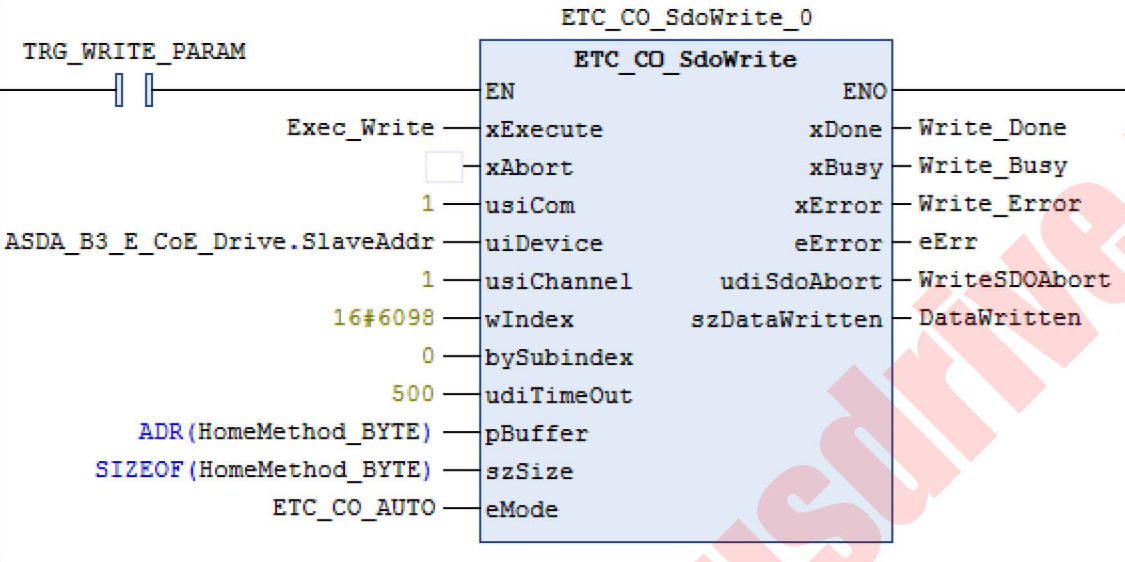




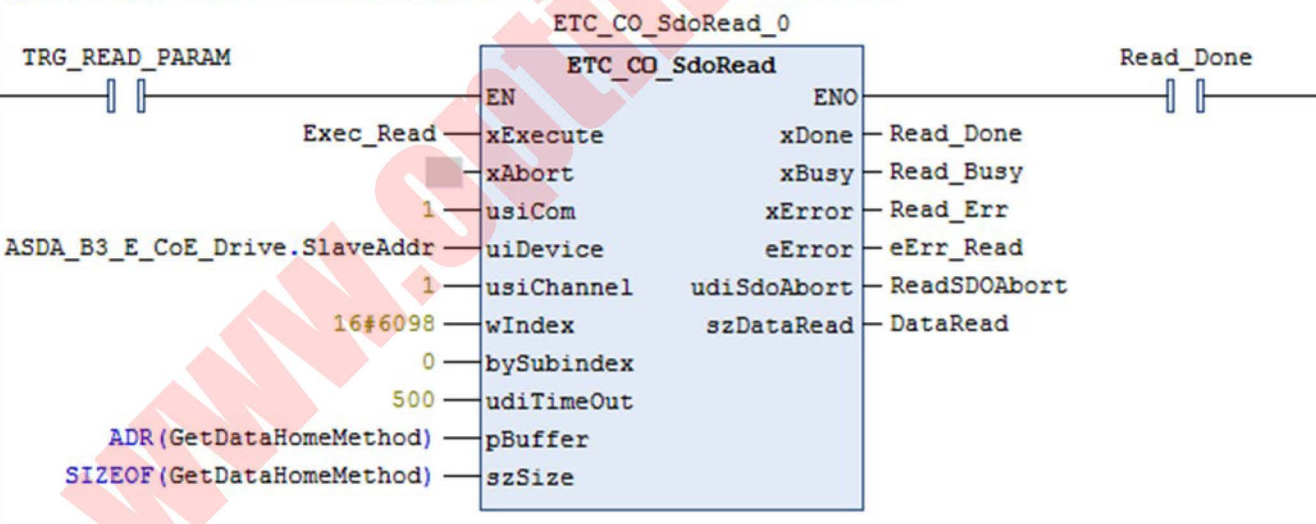
Для установки режима выхода в ноль, а также записи/чтению других параметров привода нужно использовать команды:

```
53 VAR HomeMethod_BYTE BYTE
```

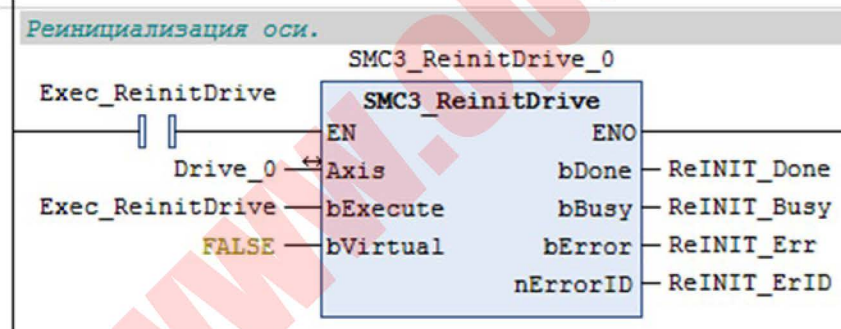
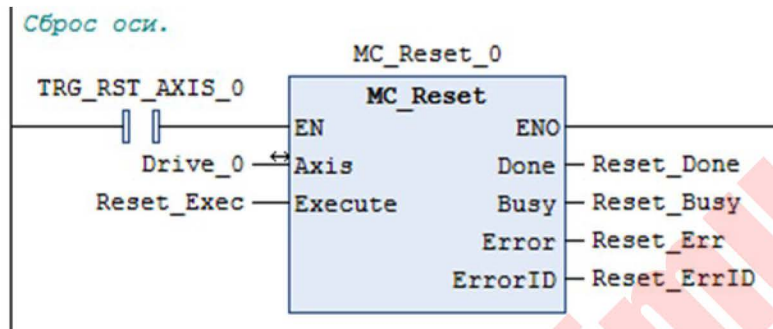
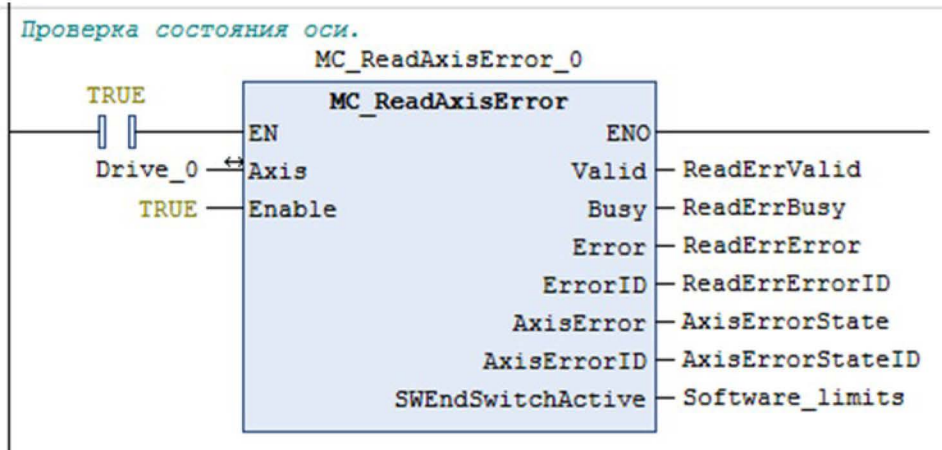
Установка режима выхода в ноль через SDO.



Чтение заданного режима выхода в ноль через SDO для проверки записи на предыдущем шаге. Данные принимаются в переменную GetDataHomeMethod типа BYTE.



Для сброса оси и реинициализации привода используются команды:



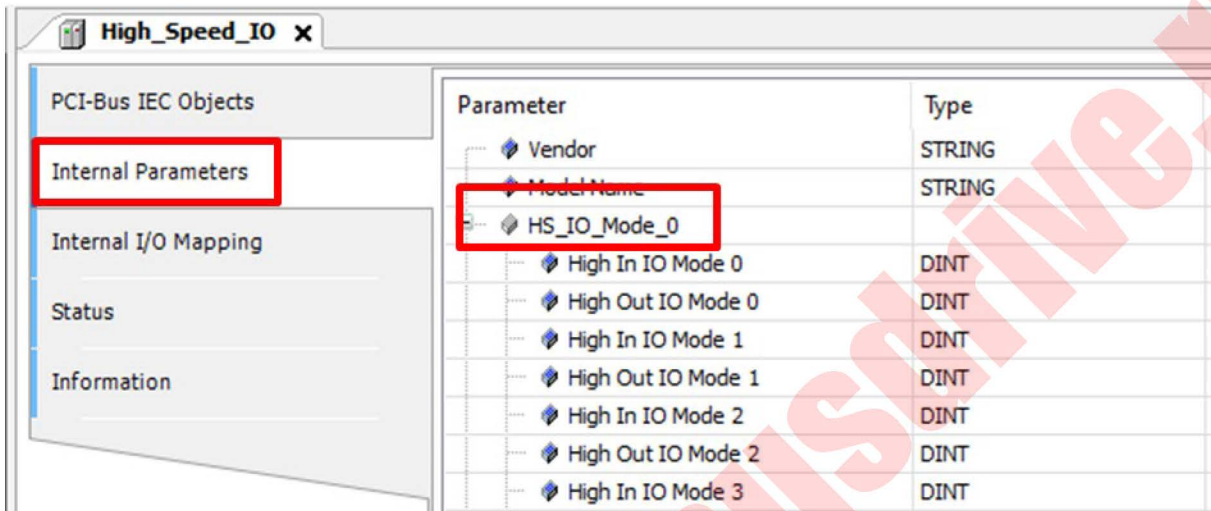
## Работа с высокоскоростными счётчиками

Контроллер имеет на борту 6 АВ-каналов высокоскоростного счёта (входы 00-11). Далее описывается последовательность настройки высокоскоростных счётчиков.

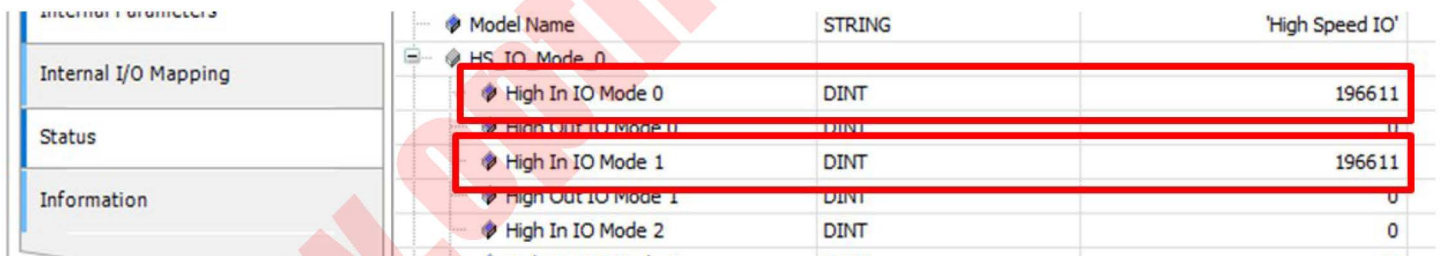
(в проект должен быть добавлен заранее объект **High\_Speed\_IO**, см. данное Руководство).

### Задание параметров работы высокоскоростных счётчиков

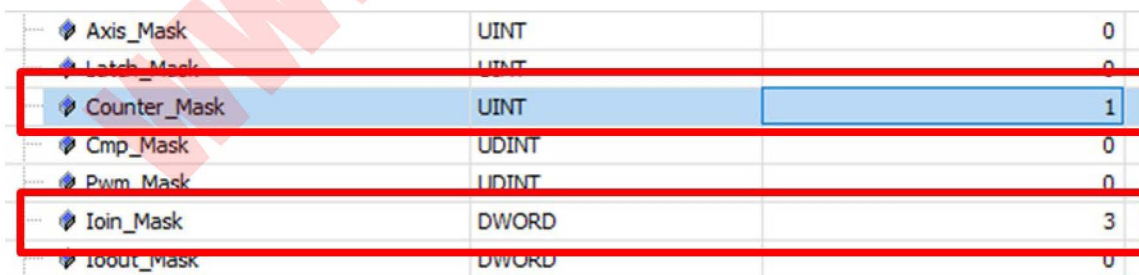
Щёлкните дважды на пункте **High Speed IO** в древе проекта и в открывшейся вкладке выберите раздел **Internal parameters**:



1) Для разрешения работы счётчика номер 0 запишите **High In IO Mode 0**=196611. Для активации других счётчиков запишите число 196611 в строку нужного счётчика.



Далее необходимо задать маски.



Для этого нужно просто сложить числа, соответствующие номерам счётчиков, которые Вы хотите использовать, согласно приведённой ниже таблицы. Например, если используется только Counter0, то **Counter\_Mask**=1 и **IoIn\_Mask** =3, если Counter0 и Counter3, то **Counter\_Mask**=1+8=9, и **IoIn\_Mask**=3+192=195)

Номер счётчика	"Counter_Mask"	"Ioin_Mask"
Counter0	20=1	20+21=3
Counter 1	21=2	22+23=12
Counter 2	22=4	24+25=48
Counter 3	23=8	26+27=192
Counter 4	24=16	28+29=768
Counter 5	25=32	210+211=3072

2) Рабочий режим счётчиков задаётся в полях типа "HS counter \*".

HS counter 1		
HS counter 2		
HS counter 3		
HS counter 4		
HS counter 5		
HS counter 6		

Раскройте нужный по номеру счётчик и найдите пункт "Counter\_SetWorkMode". Режим задаётся числом в соответствии с таблицей ниже:

Режим счёта	Значение
A/B phase (quadruple)	0
Pulse+direction	1
Single phase	2
CW/CCW	3
A/B phase	4
A/B phase(double)	5

HS counter 1			
Counter_Channel	DINT	0	0
Counter_PresetInputNum	DINT	-1	-1
Counter_SetWorkMode	UDINT	0	0
Counter_Dir	UDINT	0	0
Counter_CountMode	UDINT	0	0
Counter_MaxValue	DINT	21474...	2147483647
Counter_MinValue	DINT	-2147...	-2147483648
Counter Touch Probe Pin0	DINT	-1	-1
Counter Touch Probe Pin1	DINT	-1	-1
Counter Compare Pin	DINT	-1	-1
Counter Compare2D Pin	DINT	-1	-1
Counter A Phase	DINT	-1	-1
Counter B Phase	DINT	-1	-1
HS counter 2			

3) Выбор направления счёта  
Value =0 Вверх, -1 Вниз

Counter_Channel	DINT	0	0
Counter_PresetInputNum	DINT	-1	-1
Counter_SetWorkMode	UDINT	0	0
Counter_Dir	UDINT	0	0
Counter_CountMode	UDINT	0	0
Counter_MaxValue	DINT	21474...	2147483647
Counter_MinValue	DINT	-2147...	-2147483648
Counter Touch Probe Pin0	DINT	-1	-1
Counter Touch Probe Pin1	DINT	-1	-1
Counter Compare Pin	DINT	-1	-1
Counter Compare2D Pin	DINT	-1	-1
Counter A Phase	DINT	-1	-1
Counter B Phase	DINT	-1	-1

4) Выбор режима счёта:

Value=0 линейный счёт, -1 круговой счёт

Counter_Channel	DINT	0	0
Counter_PresetInputNum	DINT	-1	-1
Counter_SetWorkMode	UDINT	0	0
Counter_Dir	UDINT	0	0
Counter_CountMode	UDINT	0	0
Counter_MaxValue	DINT	21474...	2147483647
Counter_MinValue	DINT	-2147...	-2147483648
Counter Touch Probe Pin0	DINT	-1	-1
Counter Touch Probe Pin1	DINT	-1	-1
Counter Compare Pin	DINT	-1	-1
Counter Compare2D Pin	DINT	-1	-1
Counter A Phase	DINT	-1	-1
Counter B Phase	DINT	-1	-1

Установка фильтра

Найдите пункт "InputFilter Para" и установите нужное значение фильтра. Чем выше число фильтра, тем меньше частоты пропускаемых импульсов. Рекомендуется установить число не менее 2.

PCI-Bus IEC Objects	Parameter	Type	Value	Default Value	Unit
Internal Parameters	HS LTC 5				
Internal I/O Mapping	HS LTC 6				
Status	HS LTC 7				
Information	HS LTC 8				
	HS LTC 9				
	HS Encoder 1				
	HS Encoder 2				
	HS Encoder 3				
	InputFilter Para				
	Time_In0	UDINT	2	65535	
	Time_In1	UDINT	2	65535	
	Time_In2	UDINT	2	65535	
	Time_In3	UDINT	2	65535	
	Time_In4	UDINT	2	65535	
	Time_In5	UDINT	65535	65535	
	Time_In6	UDINT	2	65535	
	Time_In7	UDINT	2	65535	
	Time_In8	UDINT	2	65535	
	Time_In9	UDINT	2	65535	
	Time_In10	UDINT	2	65535	
	Time_In11	UDINT	2	65535	
	Time_In12	UDINT	65535	65535	

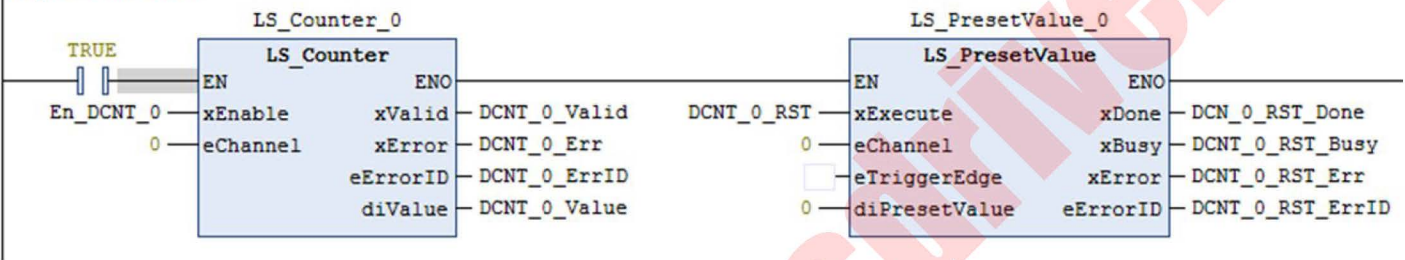
Для обращения к счётчикам в программе используйте команды:

Высокоскоростной счётчика	LS_Counter <sup>α</sup>	FB <sup>α</sup>	Инструкции высокоскоростного счётчика
	LS_PresetValue <sup>α</sup>	FB <sup>α</sup>	Команда предустановленного значения счётчика <sup>α</sup>

**LS\_Counter** - для получения текущего значения счётчика

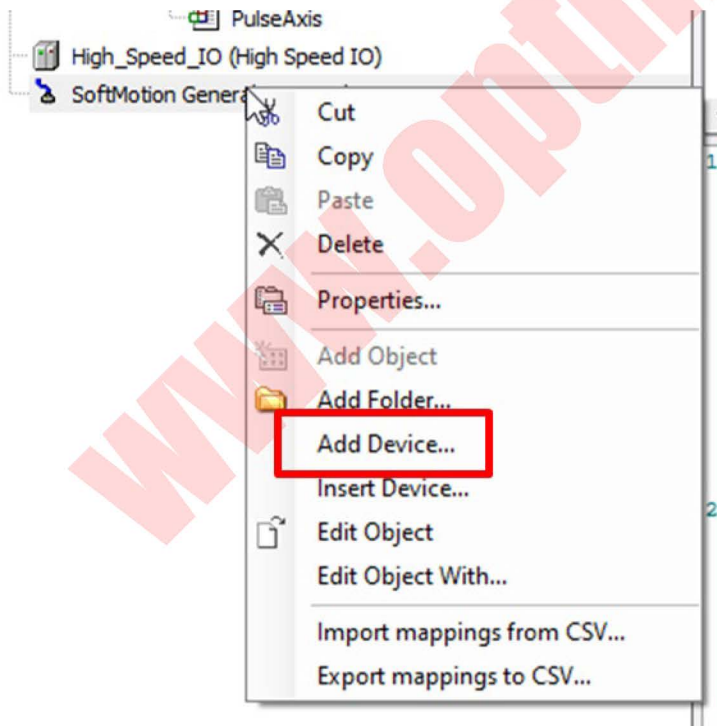
**LS\_PresetValue** - для обнуления счётчика

Чтение значения высокоскоростного счётчика 0 в режиме АВ (одиночный, как единый импульс). Входы 00/01. Сброс счётчика.

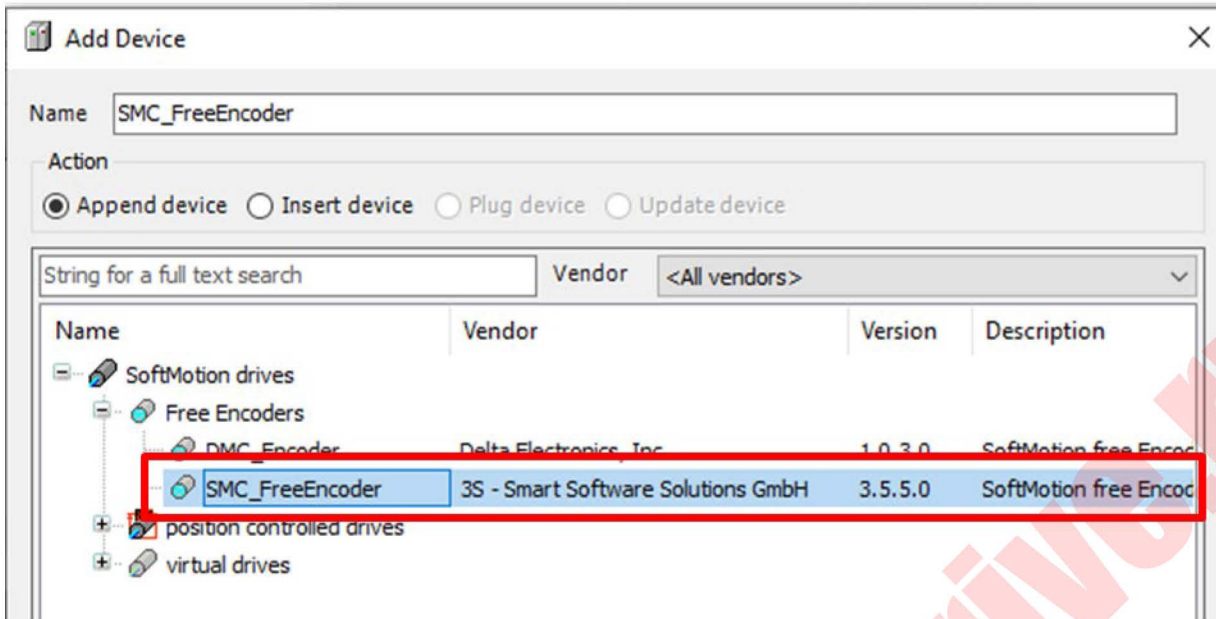


Также, высокоскоростной счётчик можно привязать к энкодерной оси и использовать в качестве Мастер-оси в различных инструкциях движения.

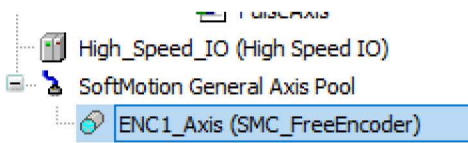
Для этого щёлкните правой кнопкой мышки на пункте **SoftMotion General Axis Pool** и в отрывшемся меню выберите пункт **Add Device**:



В открывшемся окне выберите пункт **SMC\_FreeEncoder**:

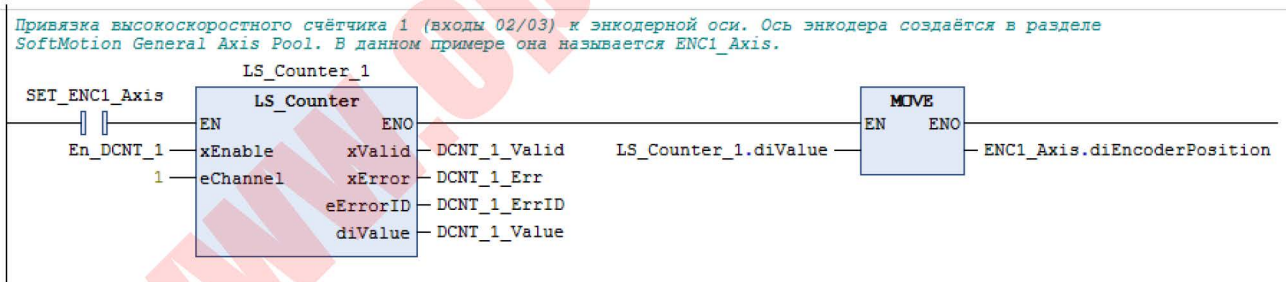


В древе проекта появится энкодерная ось, которую можно переименовать в любое удобное название, например **ENC1\_Axis**:



Высокоскоростной счётчик привязывается к энкодерной оси командой присвоения (для данного примера):

**ENC1\_Axis.diEncoderPosition:=LS\_Counter\_1.diValue;**

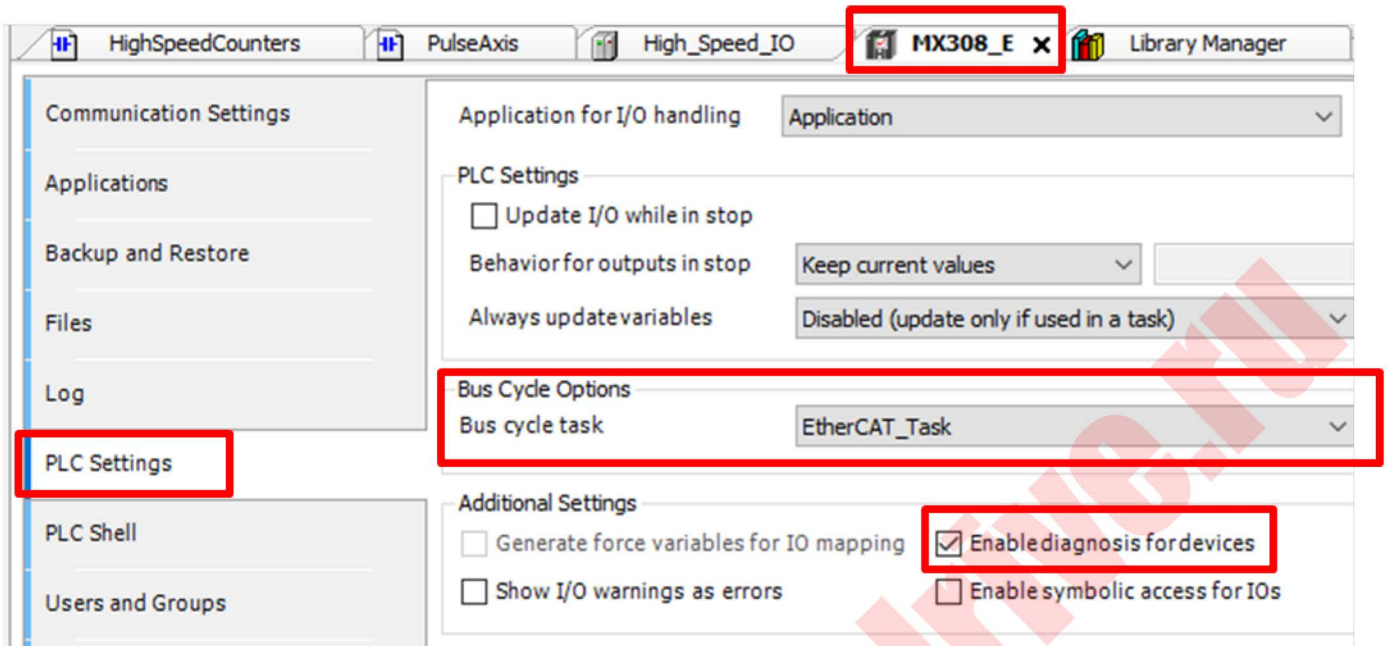


**Внимание!** POU, где используется энкодерная ось необходимо привязать к задаче EtherCAT\_Task

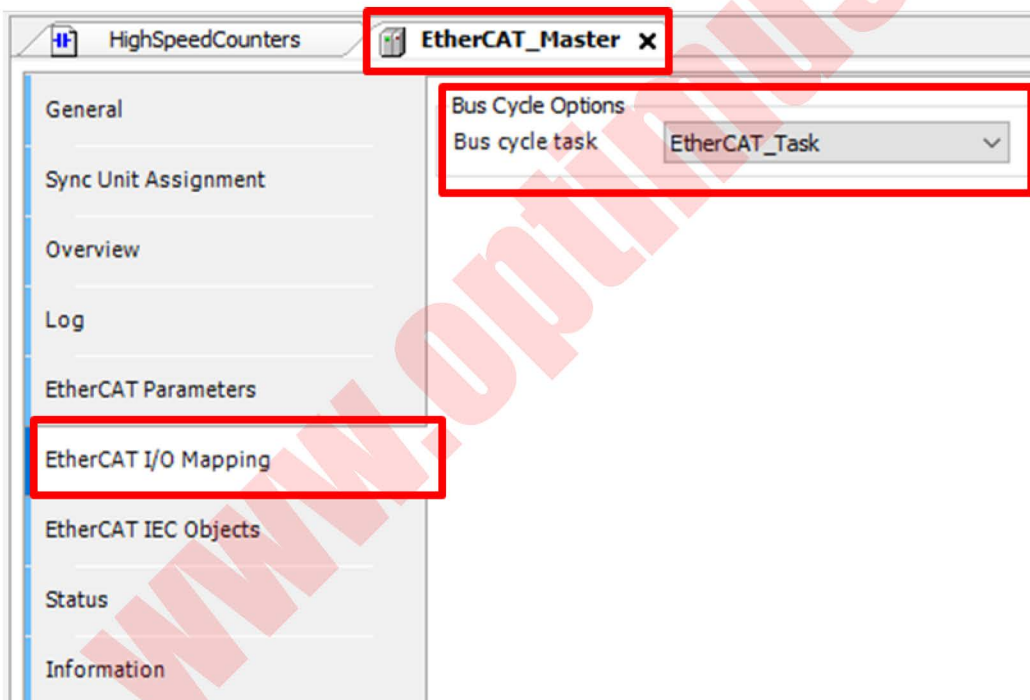
Для этого необходимо сделать ряд настроек:



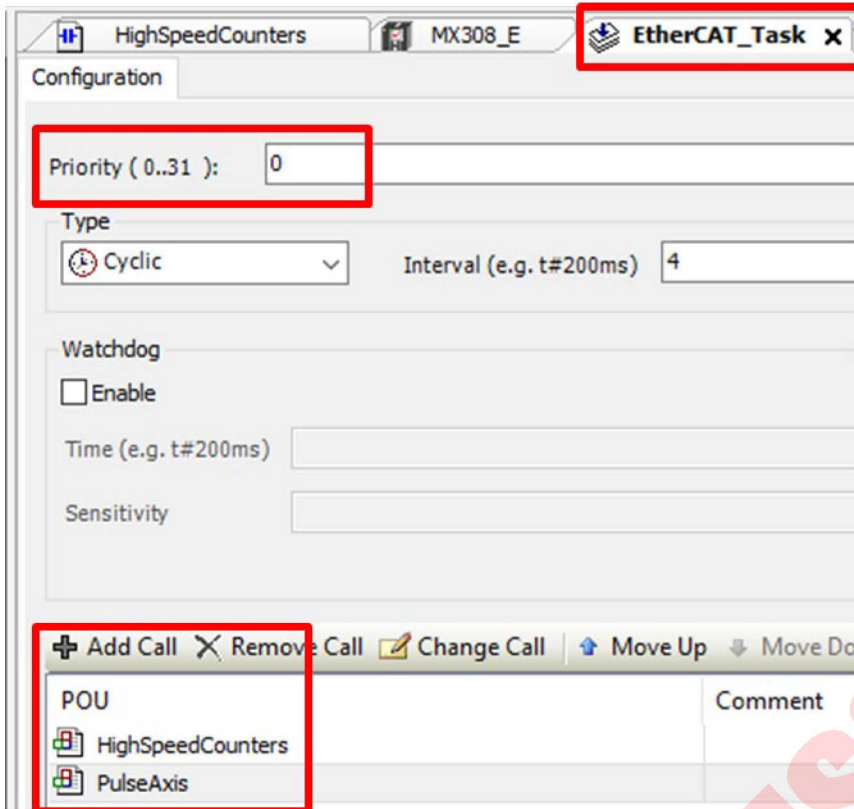
Во вкладке контроллера:



Во вкладке EtherCAT\_Master:



Во вкладке **EtherCAT\_Task** данной задаче необходимо поставить высший приоритет (0) и привязать необходимые POU:



The screenshot shows the configuration window for an EtherCAT task. The window title is "EtherCAT\_Task". The configuration is divided into several sections:

- Priority ( 0..31 ):** A text input field containing the value "0".
- Type:** A dropdown menu set to "Cyclic" and an "Interval (e.g. t#200ms)" field containing the value "4".
- Watchdog:** An "Enable" checkbox which is currently unchecked. Below it are fields for "Time (e.g. t#200ms)" and "Sensitivity", both of which are empty.
- Task List:** A table with columns "POU" and "Comment". The table contains two entries: "HighSpeedCounters" and "PulseAxis".

At the top of the configuration area, there are navigation buttons: "+ Add Call", "X Remove Call", "Change Call", "Move Up", and "Move Down".

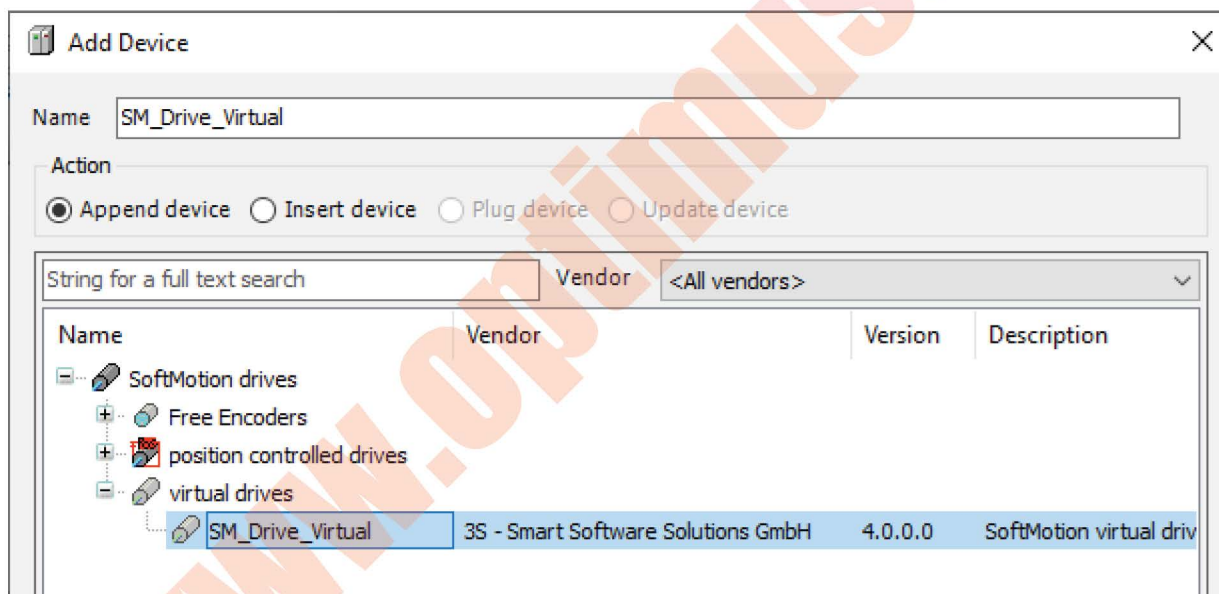
## Импульсная ось движения

Контроллеры серии MX300 имеют 12 импульсных выходов по 200 кГц, сгруппированных в 6 осей. Каждая ось соответственно занимает по 2 импульсных выхода: импульс/направление или АВ режим. Остальные 4 выхода 12-15 являются обычными.

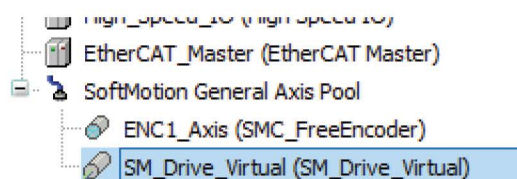
0	Импульсный выход	AXIS 0 PUL	8	Импульсный выход	AXIS 4 PUL
1	Импульсный выход	AXIS 0 DIR	9	Импульсный выход	AXIS 4 DIR
2	Импульсный выход	AXIS 1 PUL	10	Импульсный выход	AXIS 5 PUL
3	Импульсный выход	AXIS 1 DIR	11	Импульсный выход	AXIS 5 DIR
4	Импульсный выход	AXIS 2 PUL	12	Обычный выход	
5	Импульсный выход	AXIS 2 DIR	13	Обычный выход	
6	Импульсный выход	AXIS 3 PUL	14	Обычный выход	
7	Импульсный выход	AXIS 3 DIR	15	Обычный выход	

Для конфигурирования импульсной оси необходимо сначала добавить в проект адаптер входов-выходов **High\_Speed\_IO** (см. данное Руководство, раздел входов-выходов).

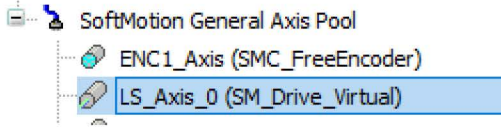
Далее добавляется Виртуальная ось. Для этого щёлкните правой кнопкой мышки на пункте **SoftMotion General Axis Pool** и в открывшемся меню выберите **Add Device**. В открывшемся окне выберите пункт **SM\_Drive\_Virtual**:



В древе проекта появится пункт:



Поменяйте название оси **SM\_Drive\_Virtual** на **LS\_Axis\_0**:



Откройте двойным щелчком левой кнопки мышки в древе проекта пункт **High\_Speed\_IO** и в открывшейся вкладке выберите пункт **Internal Parameters** и найдите настройки осей, где будет указано имя оси:

Axis_Name_0	STRING	'LS_Axis_0'	'LS_Axis_0'
Axis_Name_1	STRING	'LS_Axis_1'	'LS_Axis_1'
Axis_Name_2	STRING	'LS_Axis_2'	'LS_Axis_2'
Axis_Name_3	STRING	'LS_Axis_3'	'LS_Axis_3'
Axis_Name_4	STRING	'LS_Axis_4'	'LS_Axis_4'
Axis_Name_5	STRING	'LS_Axis_5'	'LS_Axis_5'
Axis_Mask	UINT	0	0
Latch_Mask	UINT	0	0
Counter_Mask	UINT	3	0
Cmp_Mask	UDINT	0	0
Pwm_Mask	UDINT	0	0
Ioin_Mask	DWORD	15	0
Ioout_Mask	DWORD	0	0

В разделе **HS\_IO\_Mode\_0** активируйте нужные оси путём записи числа 65537 в соответствующий пункт типа **High Out IO Mode 0**. На картинке ниже активированы оси 0 и 1 (2 оси).

Model Name	STRING	'High Speed IO'
HS_IO_Mode_0		
High In IO Mode 0	DINT	196611
High Out IO Mode 0	DINT	65537
High In IO Mode 1	DINT	196611
High Out IO Mode 1	DINT	65537
High In IO Mode 2	DINT	n

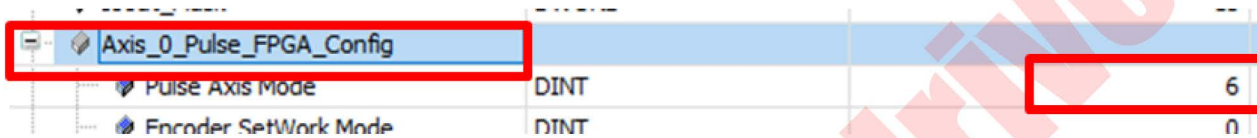
Далее необходимо задать маски.

HS_IO_Mode_0		
HS_IO_Mode_1		
Axis_Name_0	STRING	'LS_Axis_0'
Axis_Name_1	STRING	'LS_Axis_1'
Axis_Name_2	STRING	'LS_Axis_2'
Axis_Name_3	STRING	'LS_Axis_3'
Axis_Name_4	STRING	'LS_Axis_4'
Axis Name 5	STRING	'LS Axis 5'
Axis_Mask	UINT	3
Latch_Mask	UINT	0
Counter_Mask	UINT	3
Cmp_Mask	UDINT	0
Pwm_Mask	UDINT	0
Ioin_Mask	DWORD	15
Ioout_Mask	DWORD	15

Для этого нужно просто сложить числа, соответствующие номерам осей, которые Вы хотите использовать, согласно приведённой ниже таблицы. Например, если используется только Axis\_0, то "Axis\_Mask" =1 и "Ioout\_Mask" =3, если Axis\_0 и Axis\_1, то "Axis\_Mask" =1+8=9, и "Ioout\_Mask" =3+192=195)

Ось	"Axis_Mask"	"Ioout_Mask"
Axis_0	20=1	20+21=3
Axis_1	21=2	22+23=12
Axis_2	22=4	24+25=48
Axis_3	23=8	26+27=192
Axis_4	24=16	28+29=768
Axis_5	25=32	210+211=3072

В разделе Axis\_\*\_Pulse\_FPGA\_Config нужно задать режимы оси.  
Режим выдачи импульсов:



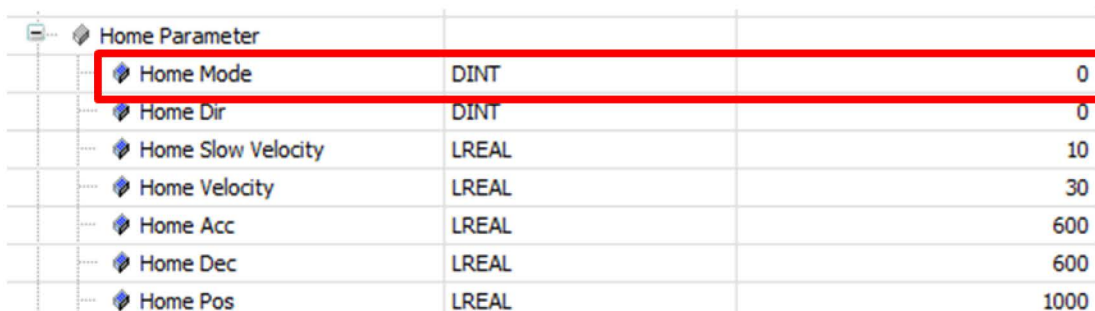
Pulse Axis Mode (Режим выдачи импульсов)	Значение
Pulse high + Direction high	0
Pulse low + Direction high	1
Pulse high + Direction low	2
Pulse low + Direction low	3
Dual pulse high	4
Dual pulse low	5
A B phase	6

Числитель/знаменатель коэффициента редукции:



Так как у контроллера максимальная частота выхода 200 кГц, то максимальное число импульсов на оборот в настройках сервопривода будет 4000 импульсов на оборот. Данное значение заносится в пункт **Axis Ratio Nume**. В пункт **Axis Ratio Denom** заносится перемещение на 1 оборот в единицах пользователя. На рисунке выше стоит 10 ед.. Это означает, что если в командах движения будет задана позиция 10, то контроллер выдаст 4000 импульсов, т.е. привод сделает 1 оборот.

Режим выхода в ноль:



Home mode (Режим выхода в ноль)	Значение
Поиск нуля в одном направлении	0
Поиск нуля в прямом направлении + поиск сигнала при обратном движении	1
Two homing processing	2
Mark homing position	3

Номер входа на ЦПУ для датчика нуля:

IsRevolveAxis	BOOL	FALSE
RevolveAxisSafeAngel	REAL	90
Home Switch Num	INT	-1
Limit Switch Num	INT	-1

Вход датчика нуля	Значение
Не используется	-1
IN0	0
IN1	1
IN2	2
IN3	3
IN4	4
IN5	5

Номер входа на ЦПУ для концевого выключателя:

IsRevolveAxis	BOOL	FALSE
RevolveAxisSafeAngel	REAL	90
Home Switch Num	INT	-1
Limit Switch Num	INT	-1

Вход концевика	Значение
Не используется	-1
IN10	10
IN11	11
IN12	12
IN13	13
IN14	14
IN15	15

В пункте **Home Switch Level** выбирается режим NO (0) или NC (1):

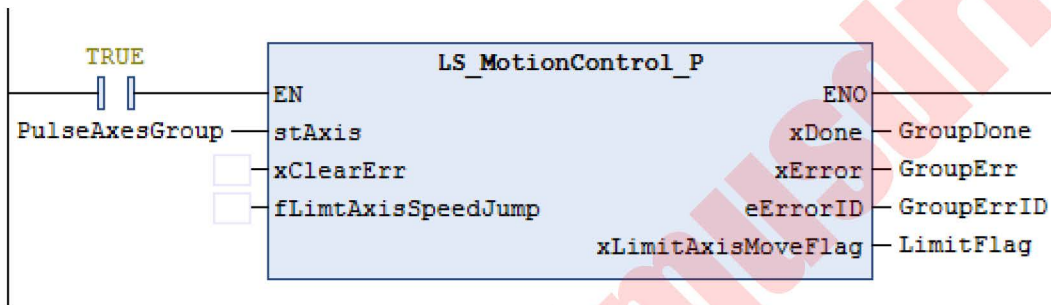
Home complete mode	INT	1
Home switch level	UINT	0
Home latch level	UINT	0
...	...	...

Параметры выхода в ноль:

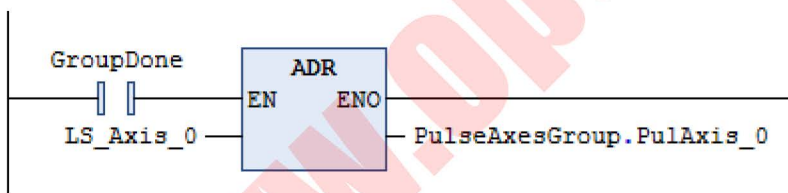
Home Parameter		
Home Mode	DINT	0
Home Dir	DINT	0
Home Slow Velocity	LREAL	10
Home Velocity	LREAL	30
Home Acc	LREAL	600
Home Dec	LREAL	600
Home Pos	LREAL	1000

В программе необходимо создать группу импульсных осей командой **LS\_MotionControl\_P**.

В примере ниже показано создание группы импульсных осей с названием PulseAxesGroup (название определяет сам пользователь). **Данная команда должна быть постоянно включена! Привязывать входные переменные не надо!** Только название группы (как в примере ниже):



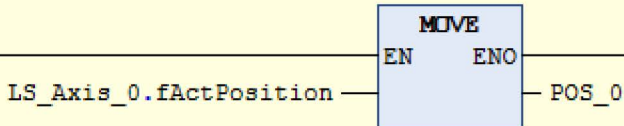
Далее необходимо командой **ADR** привязать созданные виртуальные оси к данной группе импульсных осей:



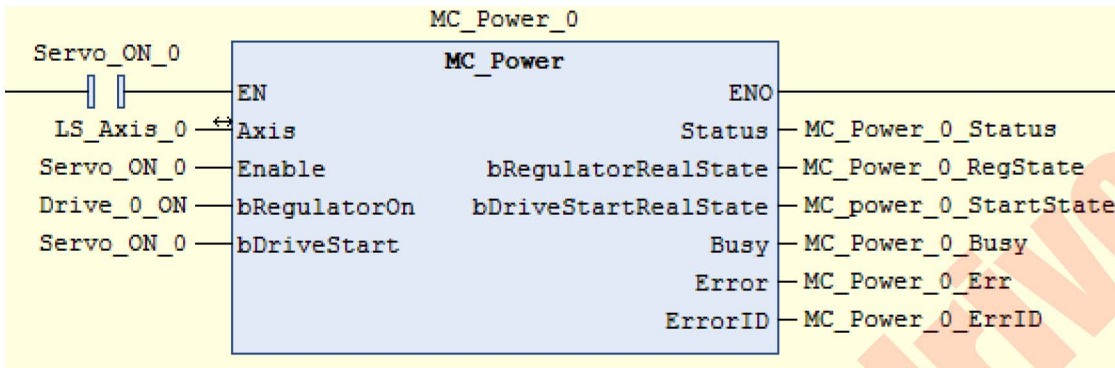
**Данная команда также должна быть постоянно включена!**

Послед указанных выше процедур импульсные оси станут доступны во всех инструкциях управления движением как и оси EtherCAT. Процедуры работы такие же. Необходимо учитывать только тот момент, что импульсным приводам необходимо подать физический сигнал Servo-ON и не будет обратной связи. Для обратной связи необходимо завести импульсные выходы привода на счётные входы контроллера. Но этого в большинстве случаев не требуется, так как серво привод имеют свою обратную связь и чётко отрабатывает задание позиции.

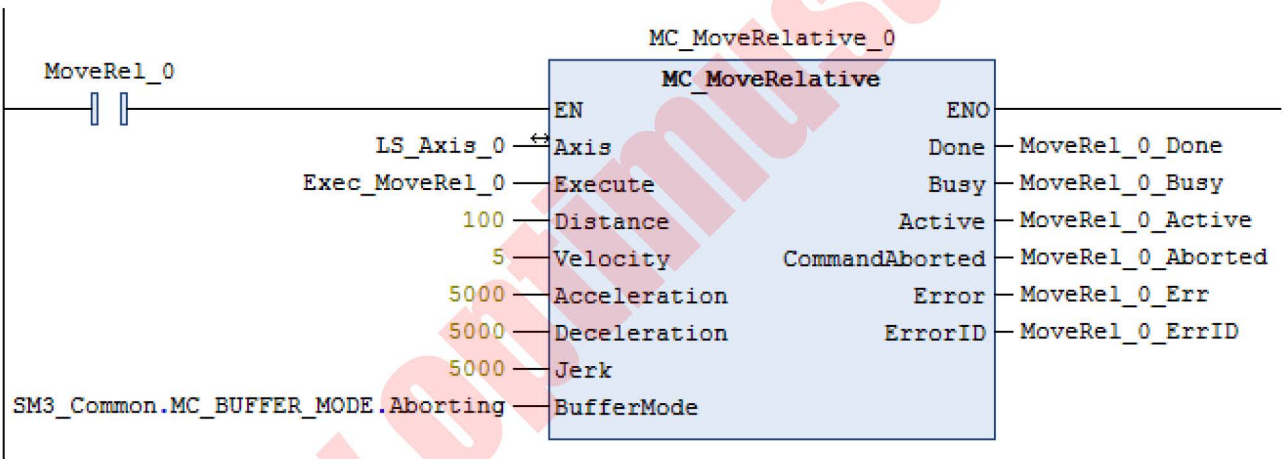
Количество выданных импульсов осью можно отслеживать через элемент структуры оси **\*\*\*.fActPosition**. Например для оси с именем **LS\_Axis\_0** обращение к данному элементу в программе будет выглядеть так: **LS\_Axis\_0.fActPosition**



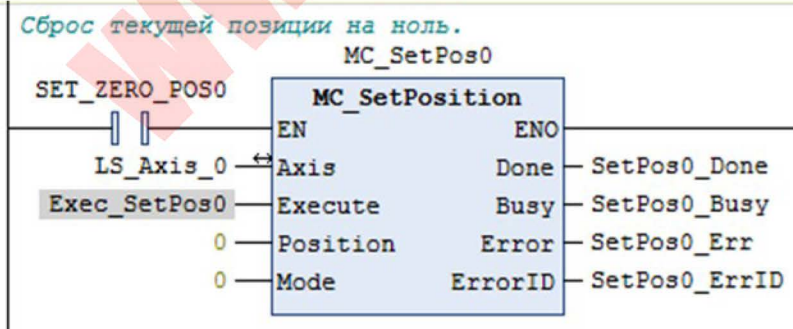
Импульсная ось также должна быть сначала запущена командой MC\_Power, а далее используются те же команды, что и EtherCAT оси.



Команда относительного перемещения:



Сброс текущей позиции оси на ноль осуществляется командой **MC\_SetPosition** с нулевым значением позиции:





## Применение штурвального энкодера в качестве мастер-оси

В данном параграфе рассматривается порядок действий для организации «ручного задатчика». В качестве Мастер-оси выступает штурвальный энкодер, а в качестве ведомой оси сервопривод Veichi SD700, управляемый по шине EtherCAT.

Штурвальный энкодер имеет фазы А и В, которые подключаются к любой высокоскоростной паре входов. Процедура организации высокоскоростных счётчиков и привязки энкодерных осей рассмотрены в параграфе «Работа с высокоскоростными счётчикам», а создание оси сервопривода рассмотрено в параграфе «Добавление в проект сервопривода».

Для данного примера используется счётчик № 5 (нумерация 0 – 5), к которому привязана энкодерная ось **Encoder5**, и ось сервопривода **Axis\_SD700**:



Как правило, штурвальные энкодеры имеют 100 импульсов на оборот. Для корректного подсчёта импульсов счётчик должен быть настроен на режим 4AB, т.е. будет приниматься 400 импульсов на 1 оборот энкодера.

HS counter 6			
Counter_Channel	DINT		5
Counter_PresetInputNum	DINT		0
Counter_SetWorkMode	UDINT		0
Counter_Dir	UDINT		0
Counter_CountMode	UDINT		0
Counter_MaxValue	DINT		2147483647
Counter_MinValue	DINT		-2147483648

В сервоприводе параметры, отвечающие за числитель и знаменатель коэффициента редукции, должны быть установлены в 1 (это параметры по умолчанию в SD700). В настройках оси надо задать количество импульсов энкодера на 1 оборот двигателя (в SD700 энкодер 17 бит, следовательно это число 131072), а также задать линейное перемещение на 1 оборот вала мотора. В нашем примере это 1000 ед. пользователя (это могут быть любые единицы длины). Остальные параметры должны быть установлены в 1.

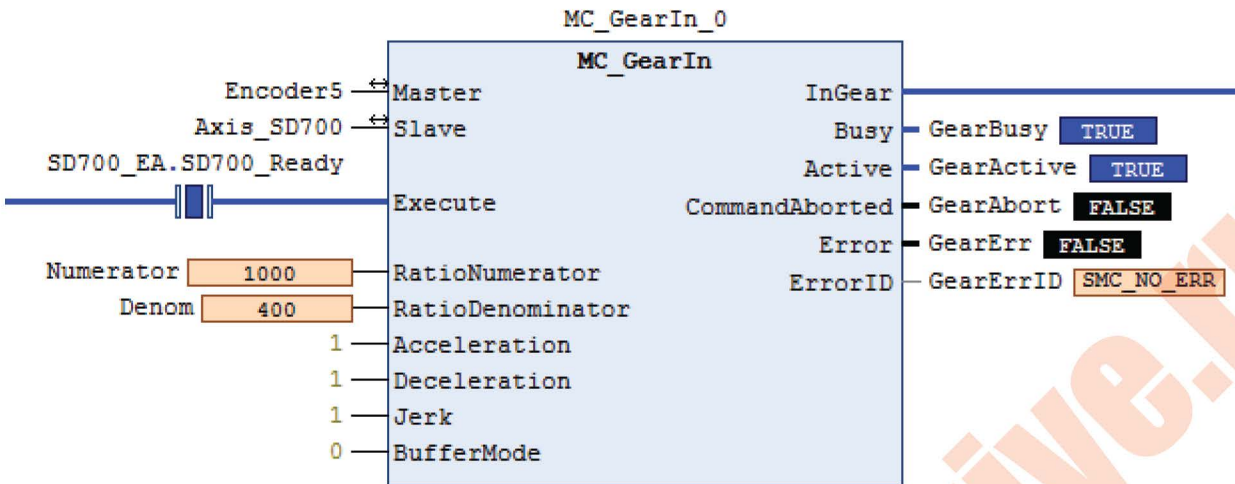
Rotary          increments <=> motor turns    

Linear          motor turns <=> gear output turns    

    gear output turns <=> units in application

Перед началом работы на природ необходимо подать команду Servo-ON (блок MC\_Power).

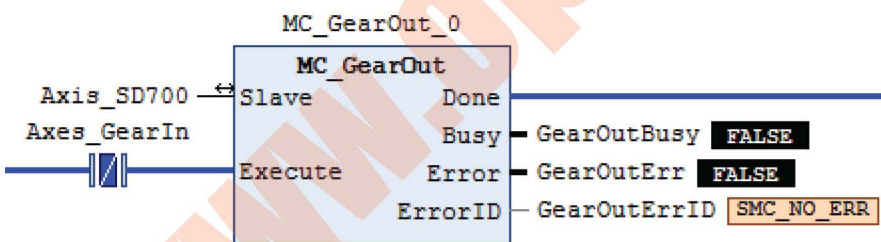
Связка Мастер ось – Ведомая ось устанавливается через команду MC\_GearIn:



В нашем примере Мастер ось (ножка Master) – **Encoder5**, Ведомая ось (ножка Slave) – **Axis\_SD700**. Числитель (Numerator) равен 1000 ед. Это из настроек оси сервопривода, расстояние в единицах пользователя.

Знаменатель (Denominator) равен 400 ед. Это количество импульсов, которое должен принять счётчик на 1 оборот энкодера. У энкодера 100 импульсов на оборот, но так как включен режим 4AB, то счётчик будет считать все передние и задние фронты, т.е. получится 400 импульсов (если у энкодера другое количество импульсов на оборот, то нужно указать количество какое есть фактически).

Расцепление осей осуществляется командой MC\_GearOut:

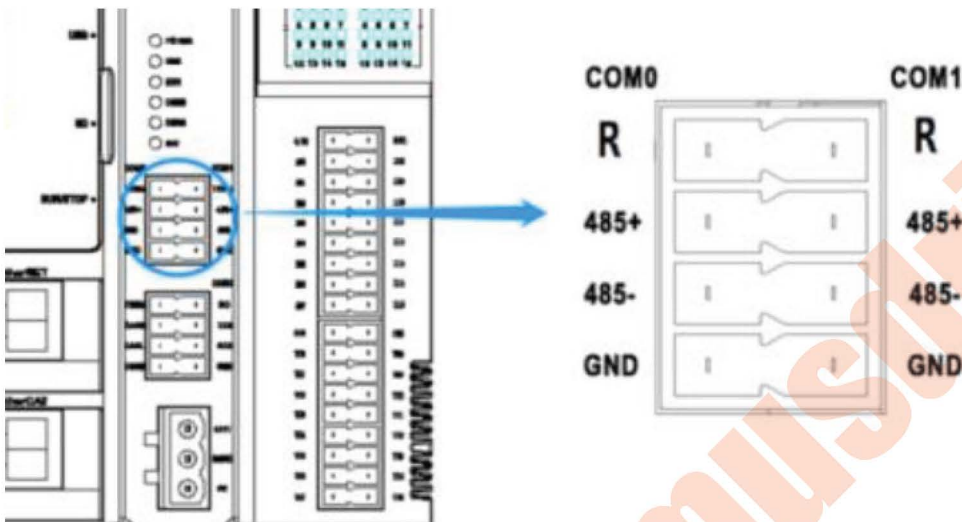


## Последовательная связь по протоколу Modbus RTU Master

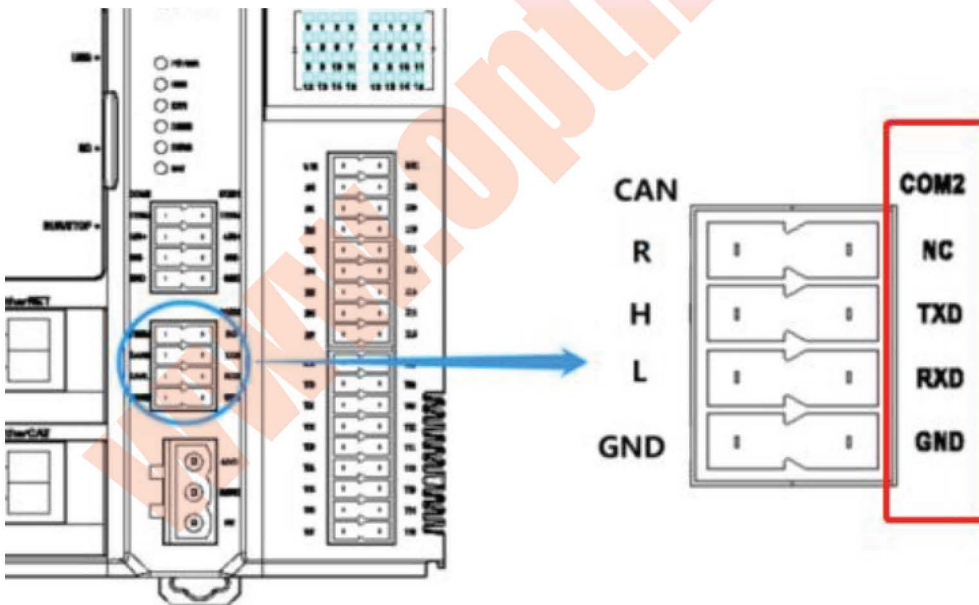
Контроллеры серии MX300 имеют на борту 1 порт RS232 и 2 порта RS485. Все три порта могут работать как в режиме Мастера, так и Ведомого. В данной главе рассмотрена работа в режиме Мастера по протоколу Modbus RTU.

Физически порты имеют на контроллере следующее расположение:

RS485:

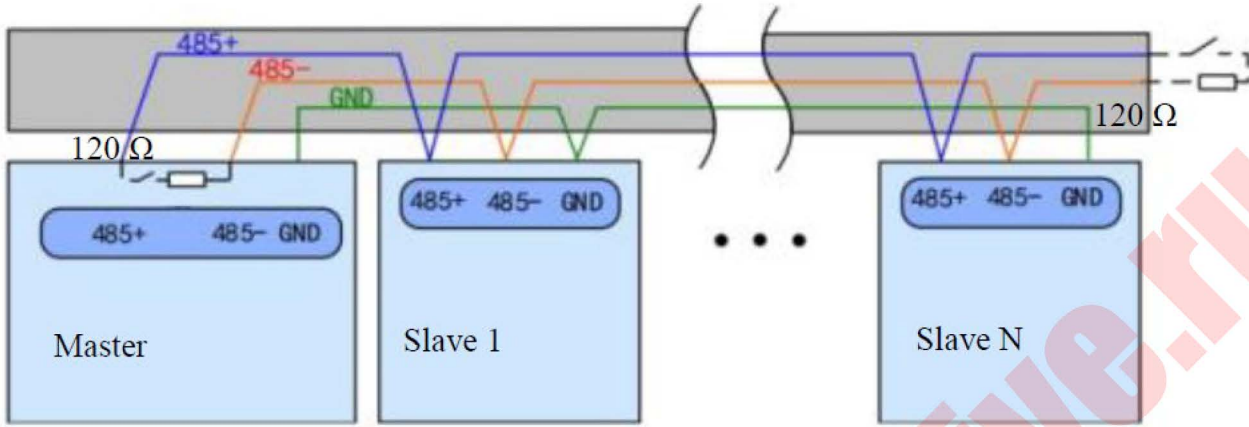


RS232:



По RS232 возможно соединение только точка-точка. Максимальная длина кабеля на скорости 9600 максимум 15 метров. На скорости 115200 – 3 метра.

По RS485 возможно подключение до 31 Ведомого. Максимальная длина кабеля на скорости 9600 до 300 метров. Далее необходимо использовать репитер. Однако, такая длина кабеля достигается только при грамотной разводке кабеля связи и использовании согласующих резисторов на концах линии, а также балластного провода для выравнивания нулевого потенциала всех станций.



У контроллеров MX300 имеется встроенный резистор 120 Ом, который подключается путём установки переключки между клеммами 485+ и R. Таким образом, мы соединяем клемму RS485+ через резистор 120 Ом на клемму RS485-.

Для организации связи по протоколу Modbus RTU Master необходимо наличие в проекте библиотеки LS\_ModbusMasterLib, которая входит в package и устанавливается вместе с ним.

```

LS_BasicModule = LS_BasicModule, 1.0.0.3 (Leadshine Technology Co.Ltd)
LS_ModbusMasterLib, 0.0.0.2 (Leadshine Technology Co.Ltd)
LS_BasicModule 1.0.0.3
LS_ModbusMasterLib 0.0.0.2
    
```

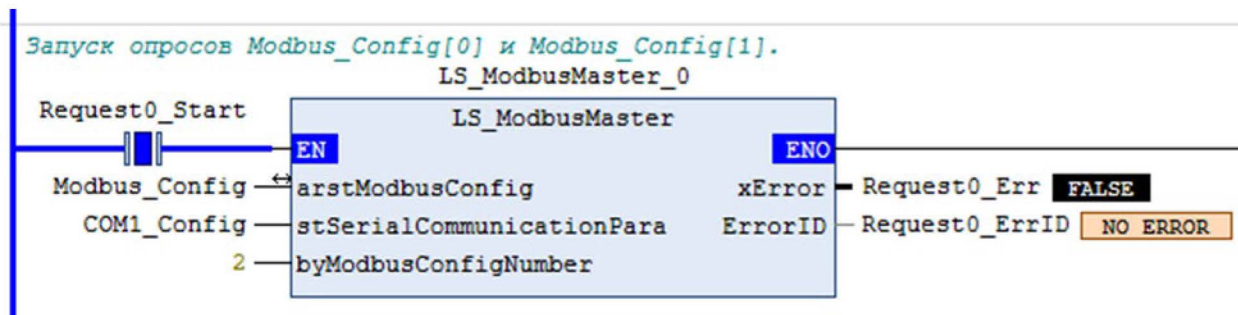
Для отправки запросов Modbus RTU используется функциональный блок **LS\_ModbusMaster**. Для каждого порта оформляется свой экземпляр блока **LS\_ModbusMaster**. И в каждый из портов можно сделать до 128 Modbus RTU запросов. Данные запросы распределяются между всею Ведомыми устройствами, подключенными к данному порту.

На языке LD экземпляра данного ФБ выглядит так:

Объявление:

LS\_ModbusMaster\_0: LS\_ModbusMaster;

Scope	Name	Address	Data type	Initialization
VAR	LS_ModbusMaster_0		LS_ModbusMaster	



Функциональный блок **LS\_ModbusMaster** имеет три входных аргумента и два выходных.

Ниже для удобства описание ножек сведено в таблицу:

Название ножки ФБ LS_ModbusMaster	Функция	Тип данных	Диапазон	Инициализация
arstModbusConfig	содержит массив структур, посредством которых конфигурируются Modbus запросы	ARRAY[0..127] OF ModbusConfig	-	-
stSerialCommunicationPara	содержит структуру, посредством которой конфигурируется последовательный порт	SerialCommunicationPara	-	-
byModbusConfigNumber	определяет количество Modbus запросов, которые блок будет последовательно отправлять в порт	BYTE	0-128	0
xError	флаг ошибки связи	BOOL	TRUE/ FALSE	FALSE
ErrorID	содержит перечисление с текстом ошибки связи	ModbusMasterErrorCode	-	0

Для запуска работы порта по протоколу Modbus RTU в режиме Мастера необходимо выполнить следующие действия:

1. Объявить экземпляр ФБ LS\_ModbusMaster
2. Заполнить структуры данных для Modbus запросов
3. Сконфигурировать порт
4. Определить количество запросов, которые будут последовательно отправляться в порт
5. Запустить опрос
6. Контролировать возникновение ошибок связи

Для оформления Modbus запросов используется структура типа **ModbusConfig**, состоящая из следующих элементов (значение заданы для примера, в реальной конфигурации будут свои):

```

usiAddress : USINT :=1 ; //Сетевой Адрес Ведомого устройства
usiFunctionCode : ModbusFuntionCode :=Read_Coils; //Функция Modbus, задаётся через ModbusFuntionCode
wOffset : WORD :=16#0000; //Адрес регистра в Ведомом устройстве в HEX, задаётся в виде 16#0000
wLength : WORD :=1; //Длина данных чтения/записи
xCycle : BOOL :=TRUE; //Метод опроса, True: циклически, False: по триггеру
uiCycleTime : UINT :=100; //Цикл опроса, ед. миллисекунды
xTrigger : BOOL ; //Сигнал триггера опроса, используется в режиме триггера
sbyRetransmissionNumber : BYTE := 3; //Количество повторных запросов
xError : BOOL := FALSE;
ErrorCode : ModbusMasterErrorCode; // Номер и текст ошибки связи, структура типа ModbusMasterErrorCode
xDone : BOOL := FALSE; //Флаг окончания передачи
arwReadData : ARRAY[0..126] OF WORD; //Буфер чтения данных (входящие данные для функций 01/02/03/04)
arwWriteData : ARRAY[0..126] OF WORD; //Буфер записи данных (исходящие данные для функций 05/06/15/16)

```

Коды функций Modbus задаются через структуру типа **ModbusFunctionCode**:

```

Read_Coils := 16#01 , //Читать катушки (дискретные выходы и маркеры)
Read_Discrete_Inputs := 16#02 , // Читать дискретные входы
Read_Holding_Registers := 16#03 ,//Читать регистры данных (16 бит)
Read_Input_Registers := 16#04 , //Читать входные регистры (16 бит)
Write_Single_Coil := 16#05 , // Запись состояния одной катушки (булевый регистр)
Write_Single_Register := 16#06 ,// Запись значения в один регистр данных (16 бит)
Write_Multiple_Coil := 15 ,// Запись состояния заданного количества катушек (булевых регистров)
Write_Multiple_Register := 16 ,//Запись значений в заданное количество регистров данных (16 бит)
    
```

Коды и название ошибок связи можно получить через структуру типа **ModbusMasterErrorCode**:

```

NO_ERROR := 0, //Ошибок нет
OPEN_SERIAL_ERROR := 100, //Ошибка открытия порта
NOT_SUPPORT_FUNCTIONCODE := 200, //Функция Modbus не поддерживается
INVALID_DATA_ADDRESS := 300, //Неправильный адрес данных
INVALID_DATA_VALUE := 400, //Значение регистра выходит за разрешённый диапазон
SLAVE_ERROR :=500, //Ошибка в Ведомом устройстве
CRC_ERROR :=600, //Ошибка контрольной суммы
INVALID_DATA_LENGTH := 700, //Неправильная длина данных
TIME_OUT := 800, //Таймаут связи
INVALID_DEVICE := 16#FFFF //Неправильное устройство
    
```

Для конфигурации запросов Modbus необходимо объявить переменную типа **ModbusConfig**

Например:

```
Modbus_Config: ARRAY [0..127] OF ModbusConfig;
```

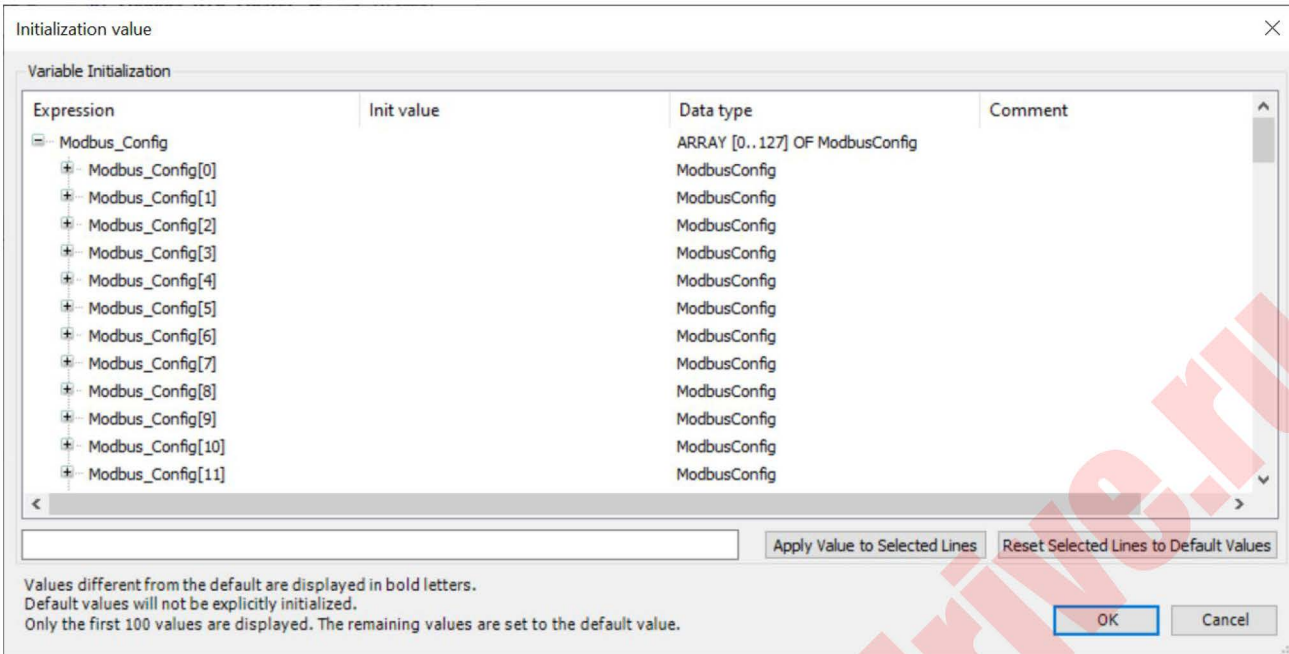
```
Modbus_Config: ARRAY [0..127] OF ModbusConfig;
```

Далее необходимо заполнить данные в структурах по количеству требуемых запросов Modbus. Это можно сделать через инициализацию, если данные не меняются, или в коде программы, если предполагается изменение данных или для наглядности.

Для заполнения структур путём инициализации переведите список переменных в табличный вид и щёлкните дважды на поле **Initialization** и нажмите на многоточие:

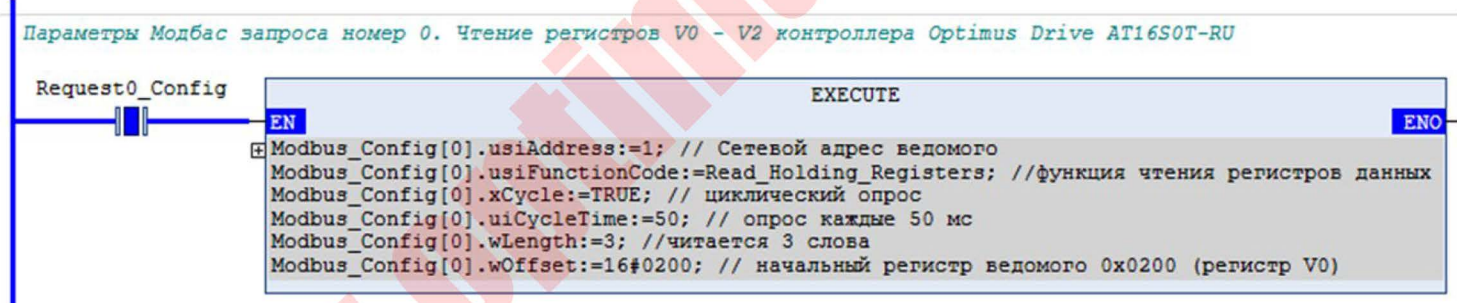
Scope	Name	Address	Data type	Initialization
VAR	Modbus_Config		ARRAY [0.. 127] OF ModbusConfig	<input type="text"/> ...

В появившемся окне выбирайте последовательно нужные структуры запросов и заполняйте данными:

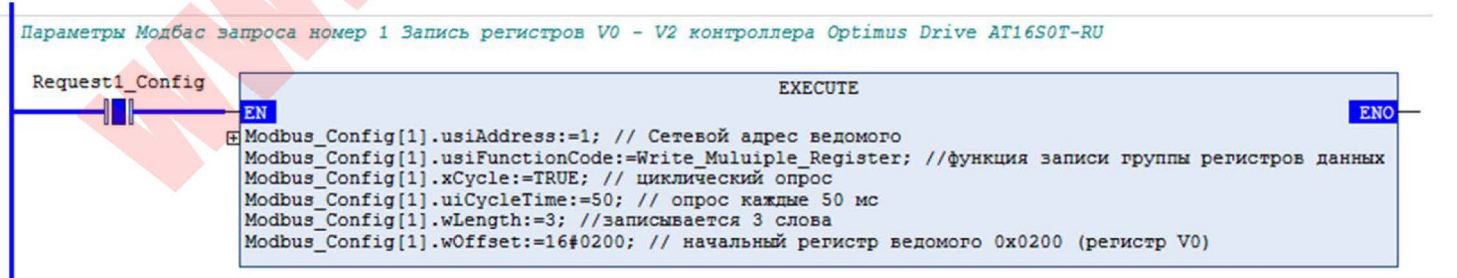


Более гибкий способ – это заполнить структуры данных в программе, используя блоки Execute. Для примера ниже приведены два блока, где заполнены данные для записи и чтения регистров V0, V1 и V2 контроллеров Optimus Drive AT16S0T-RU, который выступает в качестве Ведомого устройства:

Чтение (16#0200 – это адрес регистра V0):



Запись:



Далее необходимо сконфигурировать последовательный порт. Это также можно сделать путём инициализации в списке переменных или в тексте программы посредством блока Execute.

Настройка начинается с объявления переменной типа **SerialCommunicationPara**;

COM1\_Config: SerialCommunicationPara;

```
COM1_Config: SerialCommunicationPara;
```

Структура **SerialCommunicationPara** содержит следующие настройки порта:

```
udiPort : UDINT :=3;// Номер порта, 3: COM0(RS-485); 4: COM1(RS-485); 2: COM2(RS-232)
udiBaudrate : UDINT :=115200;//Скорость передачи
udiPARITY : COM.PARITY :=COM.PARITY.EVEN;//Проверка на чётность
udiStopBits : COM.STOPBIT :=COM.STOPBIT.ONESTOPBIT;//Количество стоповых битов
udiTimeout : UDINT :=1000;//Таймаут в миллисекундах
udiByteSize : UDINT :=8;//Количество бит данных (всегда 8, так как режим RTU)
```

Настройка порта через поле инициализации списка переменных:

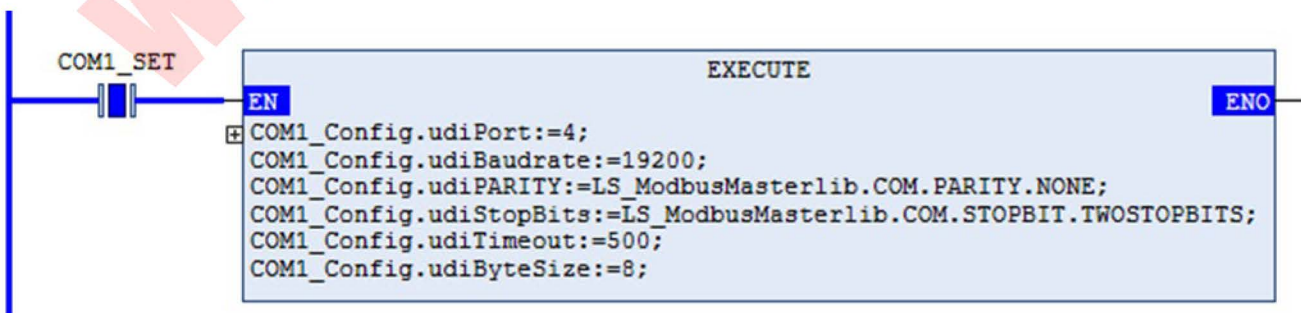
Scope	Name	Address	Data type	Initialization
VAR	COM1_Config		SerialCommunicationPara	<input type="text"/> ...

В открывшемся окне сделайте нужные настройки. Например ниже сделаны настройки для порта COM1 на протокол Modbus RTU, 19200, 8, N, 2, таймаут 500 мс для связи с контроллером Optimus Drive AT16S0T-RU:

Initialization value

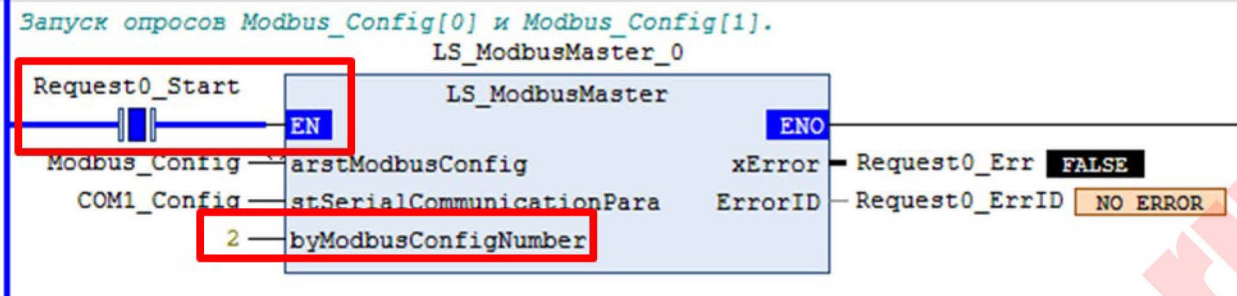
Expression	Init value	Data type
COM1_Config		SerialCommunicationPara
udiPort	4	UDINT
udiBaudrate	19200	UDINT
udiPARITY	LS_ModbusMasterlib.COM.PARITY.NONE	COM.PARITY
udiStopBits	LS_ModbusMasterlib.COM.STOPBIT.TWOSTOPBITS	COM.STOPBIT
udiTimeout	500	UDINT
udiByteSize	8	UDINT

Или сделайте настройки более гибким способом из программы контроллера путём блока Execute:





Далее необходимо задать число запросов Modbus, которые будут последовательно отправляться в порт, в нашем примере 2 запроса, и запустить инструкцию в работу путём подачи сигнала на ножку EN: (ФБ сам разводит запросы по времени)

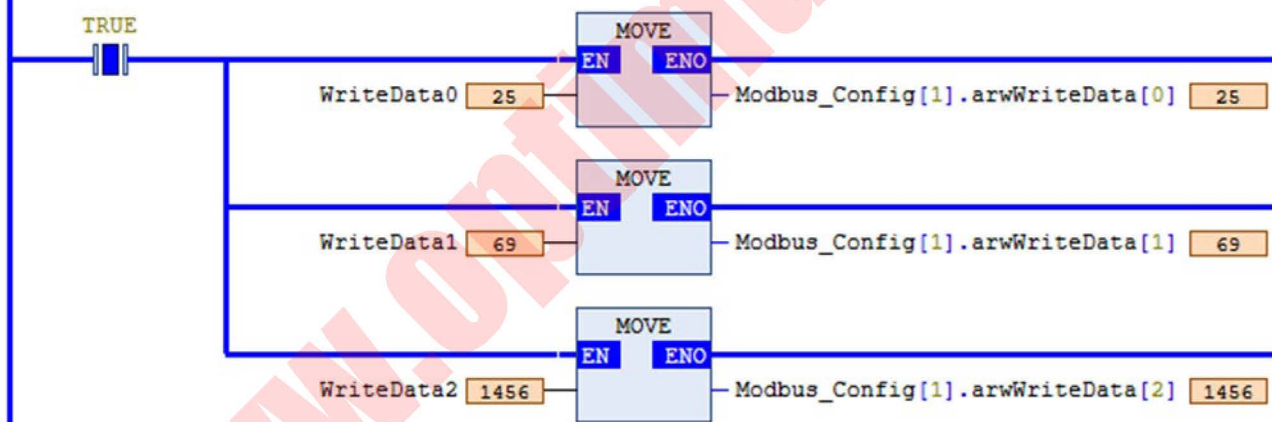


Контроллер начнёт записывать и читать данные из Ведомого устройства посредством буферов чтения и записи данных, которые являются массивами и вызываются после точки от имени соединения. Например:

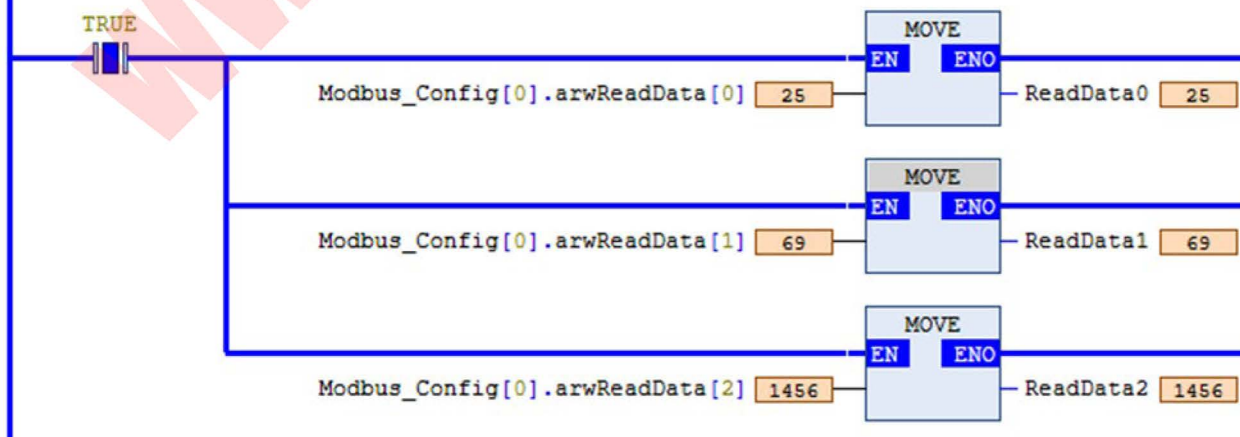
Modbus\_Config[0].arwReadData[0] - обращение к начальному регистру массива приёма данных

Modbus\_Config[1].arwWriteData[0] - обращение к начальному регистру массива отправки данных

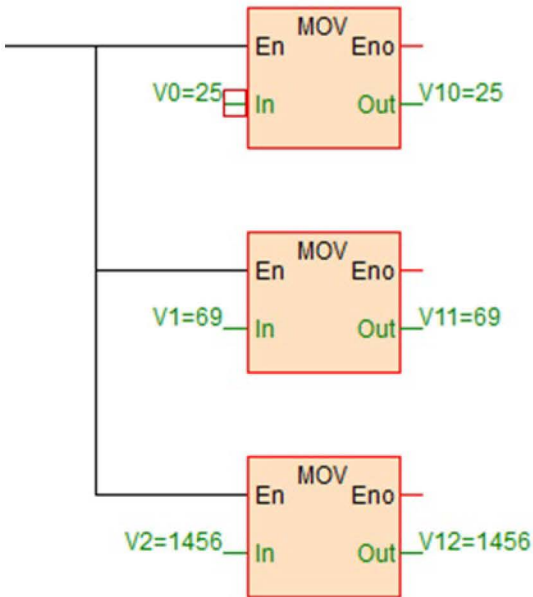
Отправка данных в регистры V0-V2 контроллера Optimus Drive AT16S0T-RU.



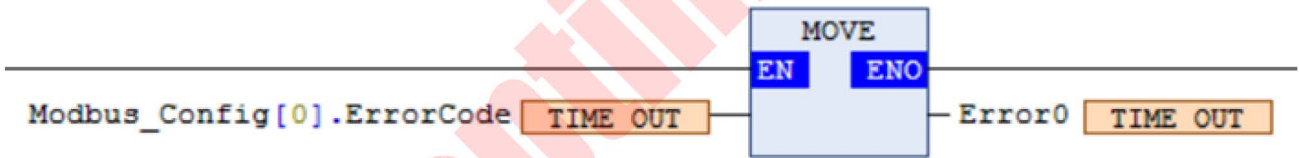
Принятые данные в регистры V0-V2 от контроллера Optimus Drive AT16S0T-RU.



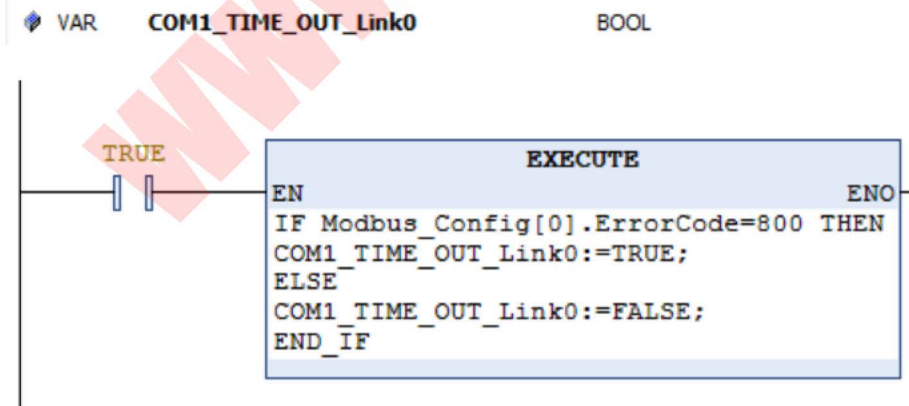
Мониторинг программы контроллера AT16S0T-RU:



Ошибки связи можно контролировать через элемент **ErrorCode** структуры **ModbusConfig**, который имеет те же номера и тексты ошибок, обозначенные в структуре **ModbusMasterErrorCode**:



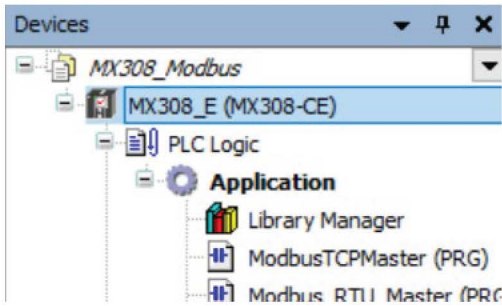
Для описания реакции на ошибки связи удобно использовать блок Execute. Например обработка таймаута связи:



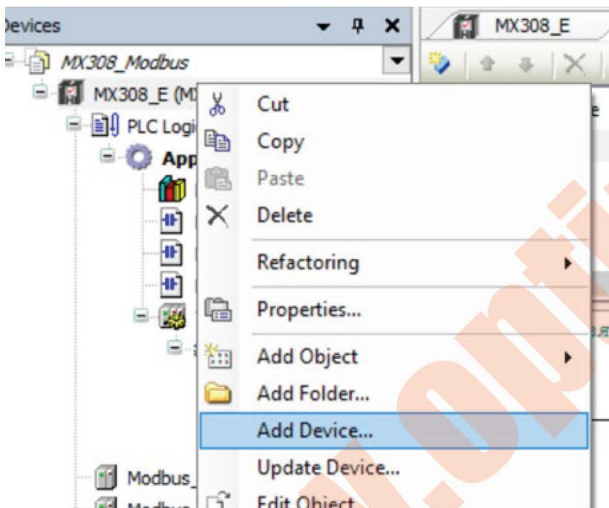
## Последовательная связь по протоколу Modbus RTU Slave

Контроллеры серии MX300 имеют на борту 1 порт RS232 и 2 порта RS485. Все три порта могут работать как в режиме Мастера, так и Водомого. В данной главе рассмотрена работа в режиме Водомого по протоколу Modbus RTU. Назначение клемм портов приведено в предыдущей главе.

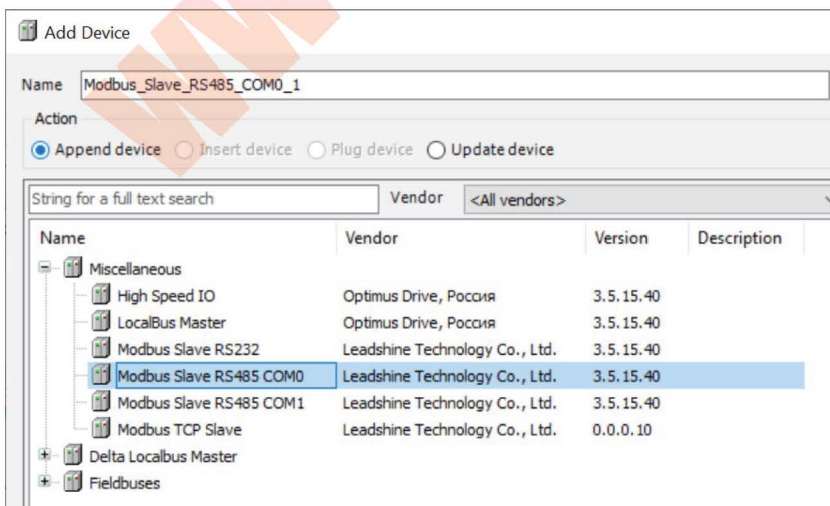
Для перевода порта последовательной связи в режим Modbus RTU Slave необходимо подключить к проекту адаптер связи типа **Modbus\_Slave\_\*\*\***. Для этого встаньте в древе проекта на пункт **Device** и нажмите правую кнопку мышки.



В появившемся меню выберите пункт **Add Device**:



В открывшемся окне выберите пункт **Miscellaneous** и адаптер нужного порта, в нашем примере COM0 RS485:

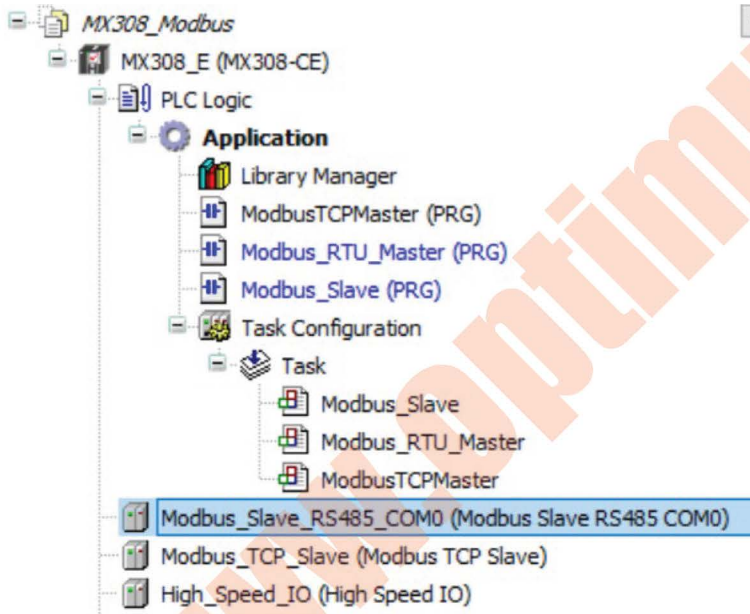


После добавления адаптера в проект и его загрузки, контроллер открывает доступ к своей памяти по протоколу Modbus RTU через соответствующий порт. Адресация будет одинаковой для всех портов. Согласно спецификации протокола Modbus, он имеет возможность читать булевы и словные регистры без обозначения типов данных в них. Поэтому таблица адресов Modbus будет выглядеть следующим образом:

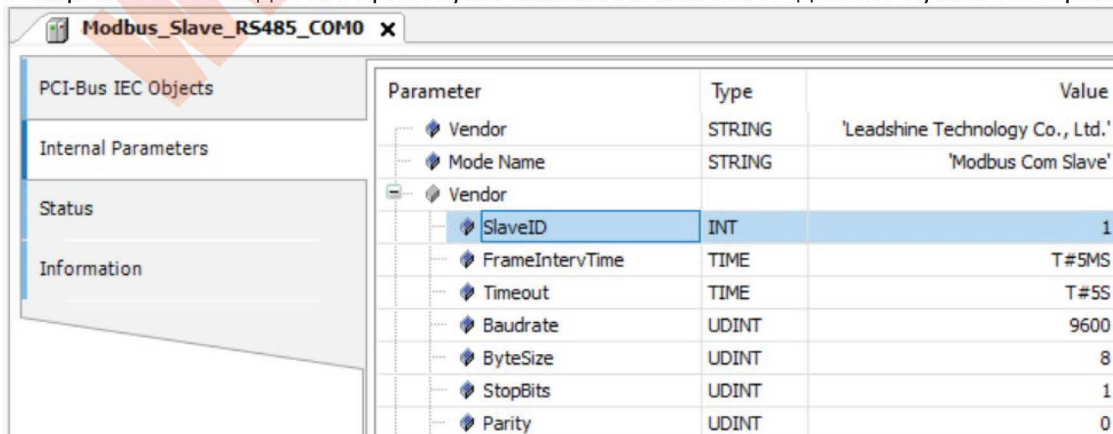
Тип	Диапазон	Функция	Инициализация	Количество
Q (выходы)	(QX0.0 ~ QX8191.7)	0x01,0x05,0x0f	0	65536
I (входы)	(IX0.0 ~ IX8191.7)	0x02	0	65536
M (данные)	%MW0~%MW65535	0x03,0x06,0x10	0	65536

Регистры типа %MW0~%MW65535 являются стандартными словами 16 бит. Адресация к битам в словах не поддерживается. Поэтому в программе биты типа %MX0.0... можно использовать, но их состояние нужно передавать как словный регистр, а потом на стороне Мастера разбирать по битам. Например, маркеры %MX0.0.. %MX0.7 и %MX1.0.. %MX1.7 будут входить в состав регистра %MW0, маркеры %MX2.0.. %MX2.7 и %MX3.0.. %MX3.7 будут входить в состав регистра %MW1 и т.д.

Для настройки скорости передачи и протокола связи щёлкните два раза на пункте адаптера порта в древе проекта:



В открывшейся вкладке выберите пункт **Internal Parameters** и сделайте нужные настройки:



После загрузки проекта в контроллер, он будет в состоянии отвечать на Modbus RTU запросы с заданной скоростью и форматом данных (9600, 8,N,1 в нашем примере).

Режим Modbus ASCII контроллерами серии MX300 не поддерживается.

Контроллеры серии MX300 могут выступать в качестве Ведомого устройства для любого стандартного Мастера Modbus RTU. Например, можно рассмотреть в связь с панелью оператора Optimus Drive VI20-070S-FE-RU.

В панели необходимо выбрать стандартный Modbus RTU драйвер:

Communication Connection

Ethernet PLC	Service	Printer	K
COM1	COM2	COM3	Remote HMI

Unused   
  Connect Device(Master)   
  Provide Servi

Manufacturer: Modbus Compatible

Device Type: Modbus RTU

Device Alias: Modbus\_RTU

Pre-set Station No.: Constant 1 Synchronize Station No.

Broadcast Station:      Master Station No.: 1

Задать протокол и во вкладке **Advance** убрать смещение адреса:

Communication Setting

Communication Type: RS485-2

Baud Rate: 9600

Data Bit: 8

Stop Bit: 1

Parity Bit: None

Reset    Advance

Protocol Timeout2: 0

Max Bit Registers: 64

Time Interval: 30

Base Address: 0

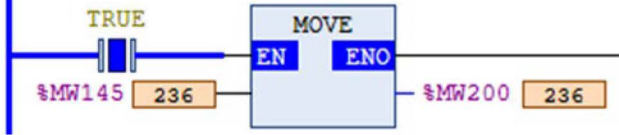
32-bit Integer: 4321

В программе контроллера создана простая программа для отображения данных, в которой задействованы следующие регистры:



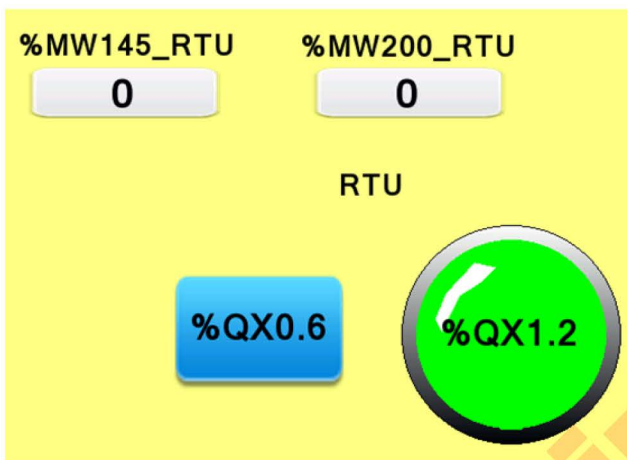
Регистр %QX0.6 - контакт, регистр %QX1.2 - выходная катушка.

Связь через Modbus RTU Slave. Порт COM0. В качестве Мастера панель оператора VI20-070S-FE-RU.

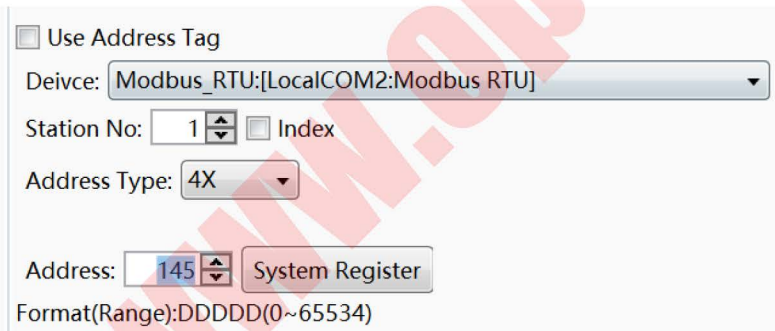


Регистр %MW145 - источник данных, регистр %MW200 - приёмник данных.

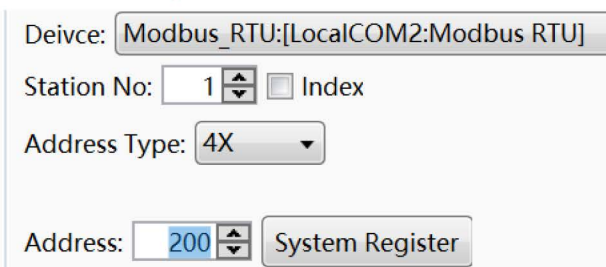
В панели оператора нарисован простой экран с четырьмя объектами соответственно: Input, Display, Button и Indicator.



В панелях используется десятичное задание адреса, поэтому обращение к регистру %MW145 будет выглядеть так:



а к регистру %MW200 так



Таким образом, нужно просто указать номер регистра %MW\*\*\*\*.

Обращение к булевой регистру %QX0.6 выглядит так:

Device: Modbus\_RTU:[LocalCOM2:Modbus RTU]  
Station No: 1  Index  
 Bit-index within a Byte Register  
Address Type: 0X  
Address: 6   
Format(Range) DDDDD(0~65534)

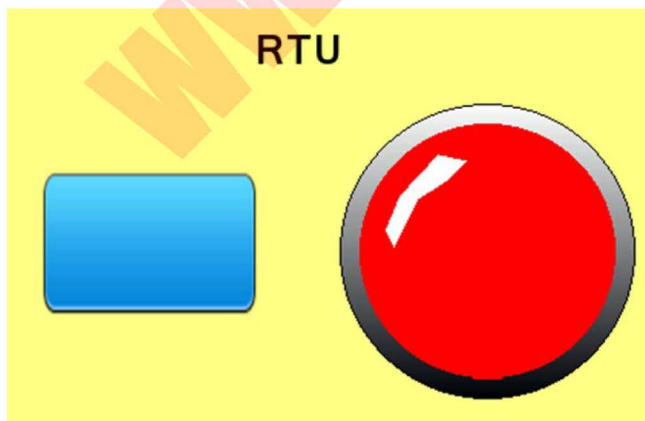
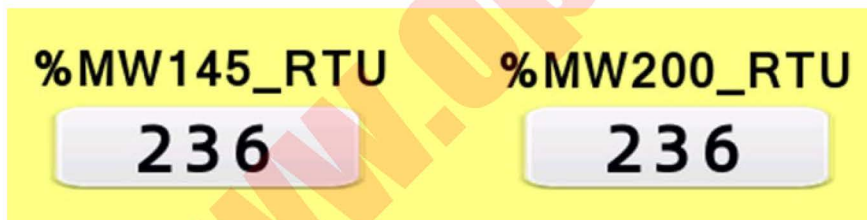
а к регистру %QX1.2 так:

Device: Modbus\_RTU:[LocalCOM2:Modbus RTU]  
Station No: 1  Index  
 Bit-index within a Byte Register  
Address Type: 0X  
Address: 10   
Format(Range):DDDDD(0~65534)

Т.е. идёт сплошная десятичная адресация. Каждый последующий регистр увеличивает адрес на 1.

%QX0.0.. %QX0.7 имеют десятичные адреса 0..7  
%QX1.0.. %QX1.7 имеют десятичные адреса 8..15  
%QX2.0.. %QX2.7 имеют десятичные адреса 16..23  
и т.д.

При начале опроса контроллера панель будет отображать состояние регистров:



## Связь по протоколу Modbus TCP Master

Контроллеры серии MX300 могут работать как в режиме Modbus TCP Client (Master), так и в режиме Modbus TCP Server (Slave), причём оба режима могут использоваться одновременно. В данной главе рассматривается организация связи контроллера с Ведомыми устройствами по протоколу Modbus TCP Client (Master).

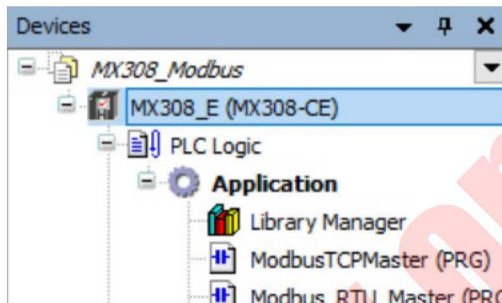
Для использования Modbus TCP Client (Master) к проекту должны быть подключены следующие библиотеки: **IoDrvEthernet**, **IoDrvModbusTCP** и **DED**.

Name	Namespace	Effective Version
CAA Device Diagnosis = CAA Device Diagnosis, 3.5.15.0 (CAA Technical Workgroup)	DED	3.5.15.0
IoDrvEthernet = IoDrvEthernet, 4.2.0.0 (3S - Smart Software Solutions GmbH)	IoDrvEthernet	4.2.0.0
IoDrvModbusTCP = IoDrvModbusTCP, 4.3.0.0 (3S - Smart Software Solutions GmbH)	IoDrvModbusTCP	4.3.0.0

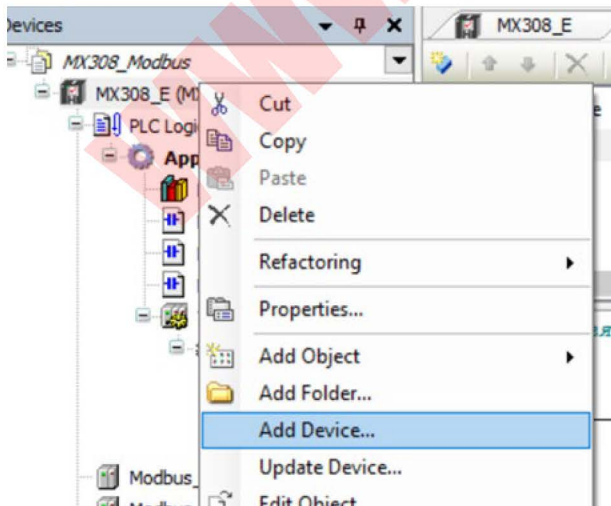
Также, для работы стандартного функционала типа переднего/заднего фронта, таймеров, счётчиков и т.п. к проекту должна быть подключена библиотека **Standard**, а для использования расхожего функционала типа фликеров и т.п. библиотека **Util**.

Standard = Standard, 3.5.18.0 (System)	Standard	3.5.18.0	
Util = Util, 3.5.18.0 (System)	Util	3.5.18.0	

Для начала работы через порт Ethernet необходимо сначала добавить в проект устройство типа **Ethernet**. Для этого встаньте на пункт **Device** и щёлкните правой кнопкой мышки:

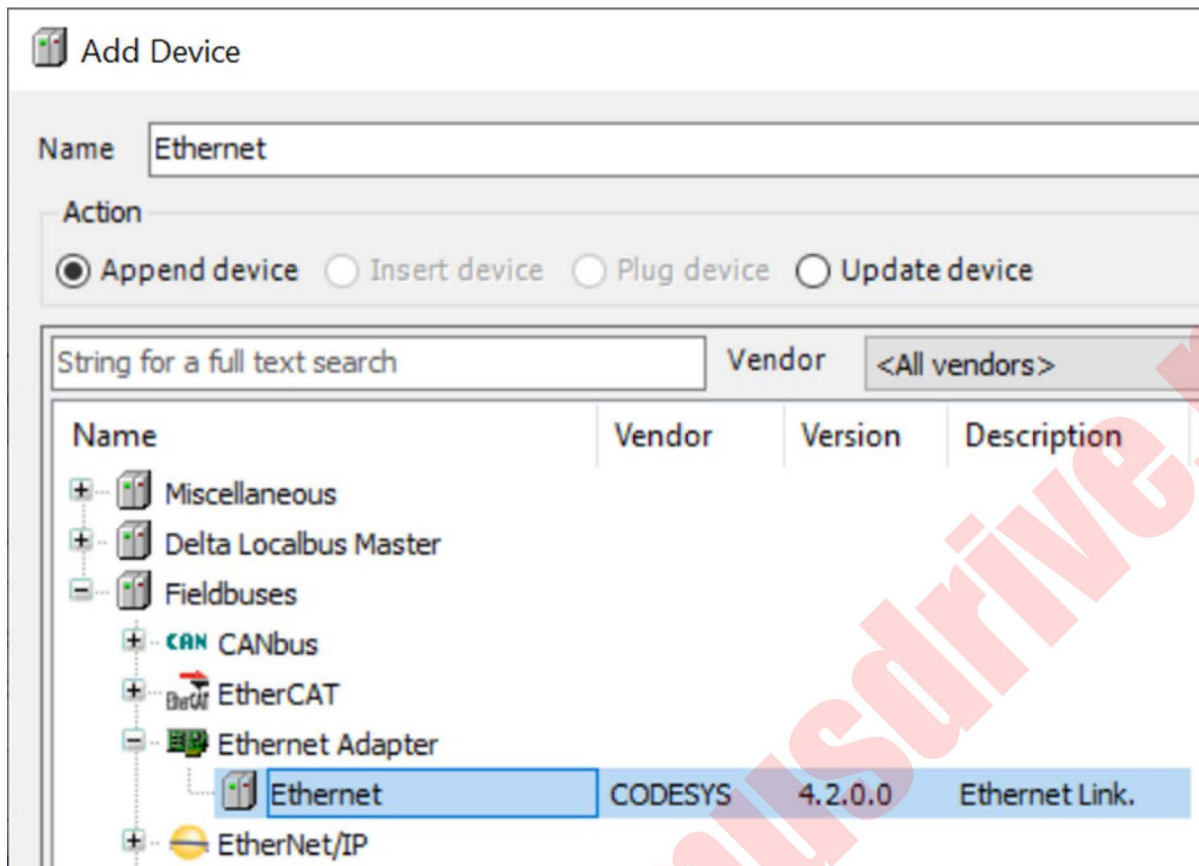


В появившемся меню выберите пункт **Add Device**:

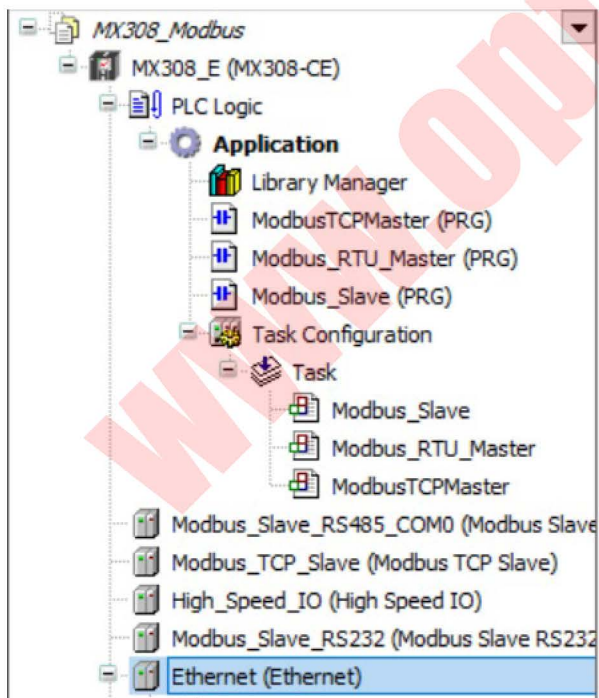




В открывшемся окне выберите пункт **Fieldbuses – Ethernet**:

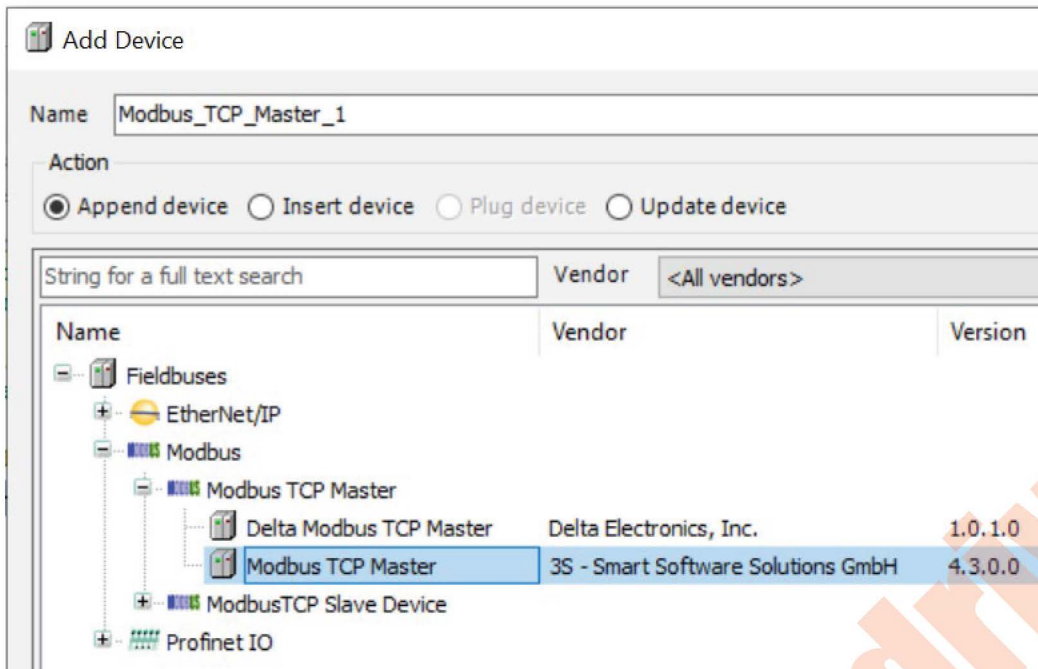


В древе проекта появится пункт **Ethernet**:

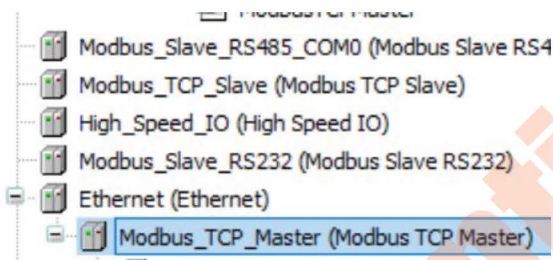


Щёлкните правой кнопкой мышки на пункте Ethernet и в открывшемся меню выберите пункт **Add Device**.

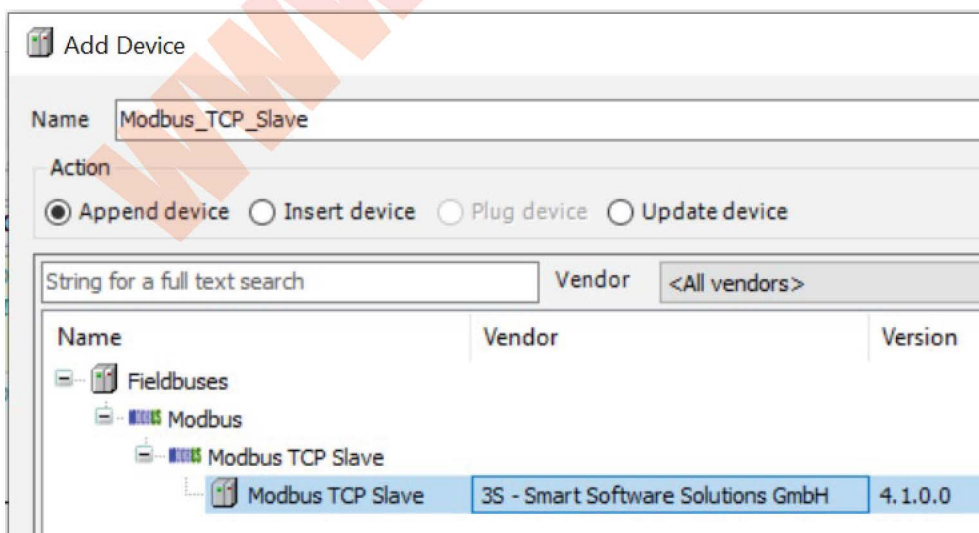
В открывшемся окне выберите пункт **Fieldbuses – Modbus – Modbus TCP Master**:



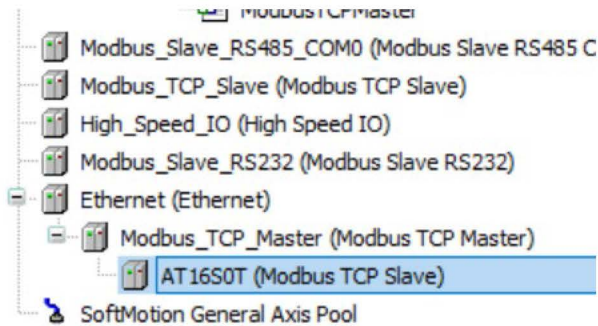
В древе проекта появится пункт **Modbus TCP Master**:



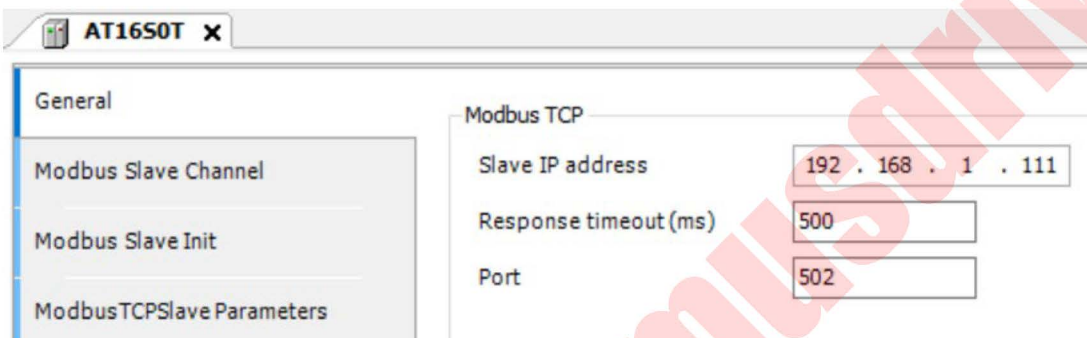
Контроллер готов к работе по протоколу Modbus TCP Client (Master) и осталось добавить линки, т.е. конкретные ведомые устройства, которые контроллер будет опрашивать. Для этого встаньте мышкой на пункт **Modbus TCP Master** и щёлкните правой кнопкой мышки и в открывшемся меню выберите пункт **Add Device**. В открывшемся окне выберите **Fieldbuses – Modbus – Modbus TCP Slave**:



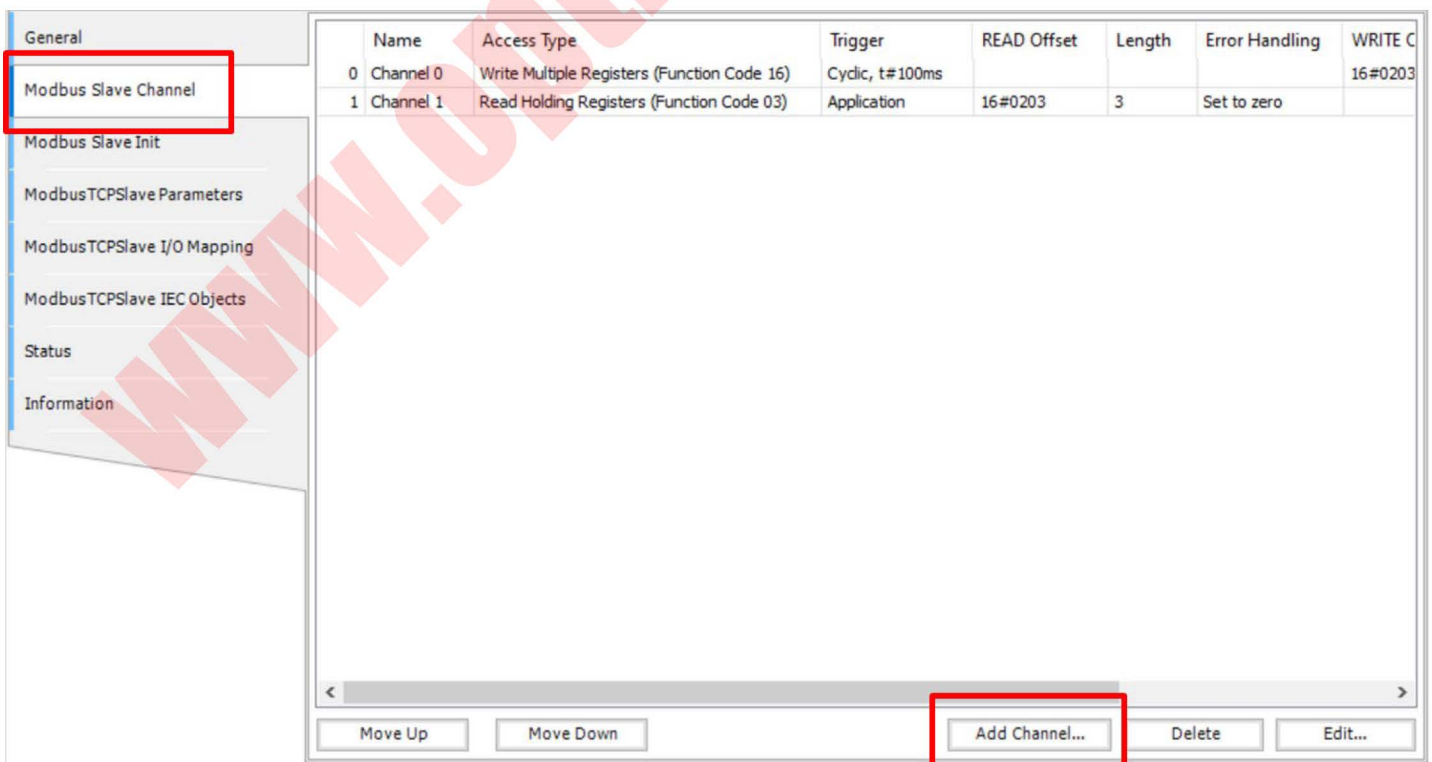
У Вас в древе проекта появится пункт с Ведомым устройством, которому можно сразу дать имя. В нашем примере в качестве Ведомого будет использоваться контроллер Optimus Drive AT16S0T-RU



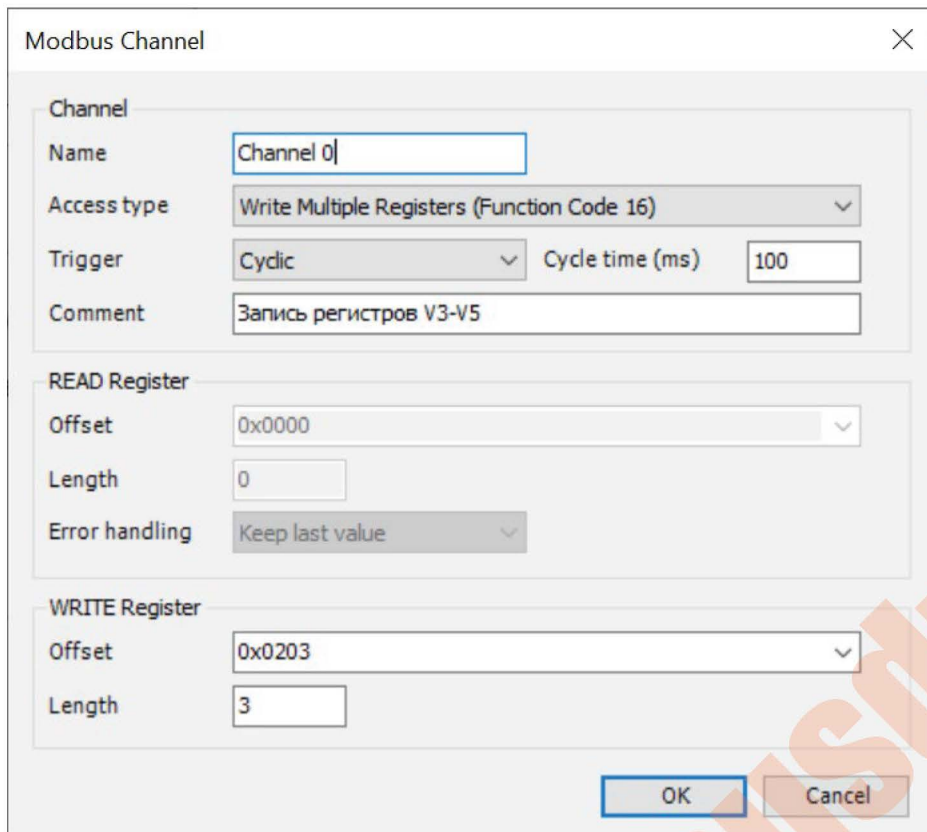
Щёлкните дважды левой кнопкой мышки, откроется вкладка настройки обмена с Ведомым. В пункте **General** установите IP адрес Ведомого (в нашем примере 192.168.1.111):



Далее во вкладке **Modbus Slave Channel** необходимо создать нужное количество запросов. Откройте пункт **Modbus Slave Channel** и для оформления запроса нажмите кнопку **Add Channel**:



Открывшееся окно содержит следующие настройки:



**Name** – Имя запроса (латинские буквы и цифры)

**Access Type** – Код функции Modbus

**Trigger** – Метод отправки запросов. **Cyclic** – циклически в автоматическом режиме, **Rising Edge** – запрос отправляется по переднему фронту назначенного регистра. Регистр можно посмотреть во вкладке **ModbusTCP Slave I/O Mapping**. И третий метод – **Application**. Позволяет управлять запросами из программы контроллера посредством ФБ **ModbusChannel**.

**Cyclic Time** – Время цикла опроса в миллисекундах. Для метода **Trigger = Cyclic**

**Comment** – Комментарии к запросу (справочная информация, к самому запросу отношения не имеет)

**READ Register** – блок, в котором оформляется чтение данных из Ведомого

**Offset** – Адрес регистра Ведомого, задаётся в HEX записью вида 0x0000

**Length** – Длина данных для чтения

**Error Handling** – Выбор сохранять или не сохранять состояние регистров при потере связи

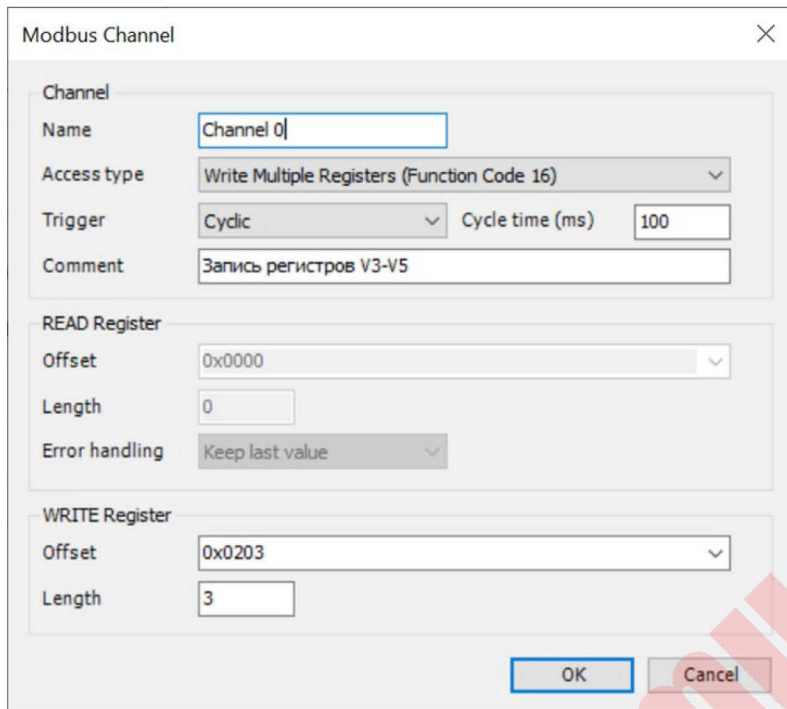
**WRITE Register** – блок, в котором оформляется запись данных в Ведомого

**Offset** – Адрес регистра Ведомого, задаётся в HEX записью вида 0x0000

**Length** – Длина данных для записи

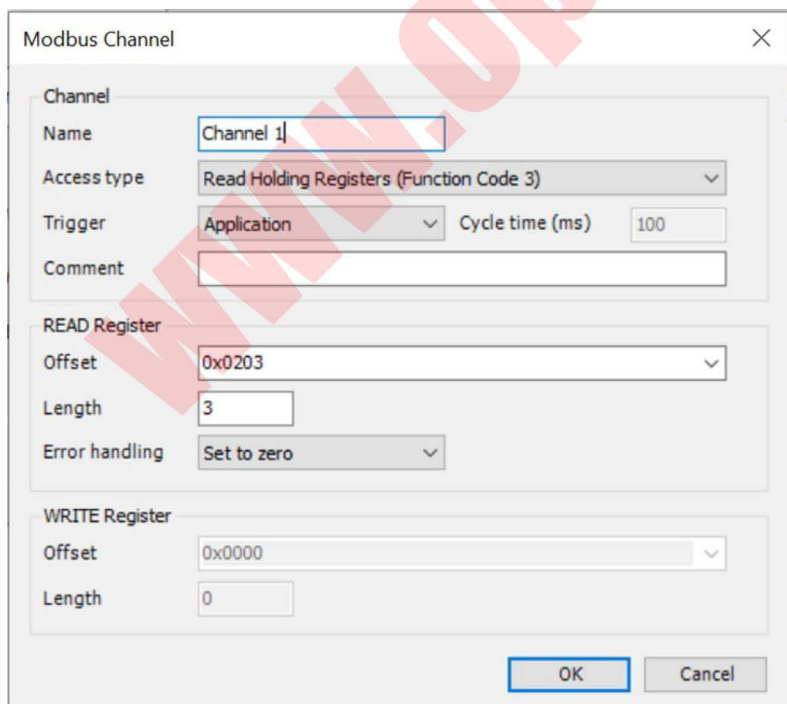
Ниже для примера оформлено два запроса Channel\_0 и Channel\_1 для записи и чтения регистров V3-V5 контроллера Optimus Drive AT16S0T-RU

Запись (адрес регистра V3 – 0x0203). Имя – Channel\_0, Команда 16 (групповая запись регистров), метод опроса – циклический с тактом в 100 мс, Начальный регистр для записи 0x0203, количество последовательно записываемых регистров 3 штуки.



The screenshot shows the 'Modbus Channel' configuration dialog for 'Channel 0'. The 'Channel Name' is 'Channel 0'. The 'Access type' is 'Write Multiple Registers (Function Code 16)'. The 'Trigger' is 'Cyclic' with a 'Cycle time (ms)' of 100. The 'Comment' is 'Запись регистров V3-V5'. The 'READ Register' section has an 'Offset' of 0x0000 and 'Length' of 0. The 'WRITE Register' section has an 'Offset' of 0x0203 and 'Length' of 3. 'OK' and 'Cancel' buttons are at the bottom.

Чтение. Имя – Channel\_1, Команда 3 (групповое чтение регистров), метод опроса – Application, из программы функциональным блоком ModbusChannel, Начальный регистр для чтения 0x0203, количество последовательно считываемых регистров 3 штуки, при обрыве связи регистры сбрасываются в ноль (Set to zero).



The screenshot shows the 'Modbus Channel' configuration dialog for 'Channel 1'. The 'Channel Name' is 'Channel 1'. The 'Access type' is 'Read Holding Registers (Function Code 3)'. The 'Trigger' is 'Application' with a 'Cycle time (ms)' of 100. The 'Comment' is empty. The 'READ Register' section has an 'Offset' of 0x0203 and 'Length' of 3. The 'Error handling' is 'Set to zero'. The 'WRITE Register' section has an 'Offset' of 0x0000 and 'Length' of 0. 'OK' and 'Cancel' buttons are at the bottom.

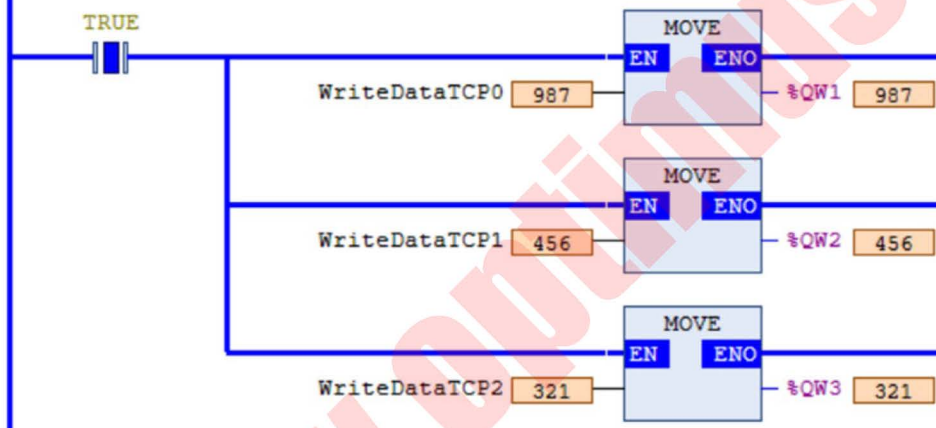
После создания каналов опроса система автоматически выделит под них регистры в соответствии с заявленным количеством регистров в Ведомом устройстве. В нашем примере будет 3 регистра на запись и 3 на чтение. Посмотреть можно в пункте **ModbusTCP Slave IEC Objects**:

Variable	Mapping	Channel	Address	Type
		Channel 0	%QW1	ARRAY [0..2] OF WORD
		Channel 0[0]	%QW1	WORD
		Channel 0[1]	%QW2	WORD
		Channel 0[2]	%QW3	WORD
		Channel 1	%IW1	ARRAY [0..2] OF WORD
		Channel 1[0]	%IW1	WORD
		Channel 1[1]	%IW2	WORD
		Channel 1[2]	%IW3	WORD

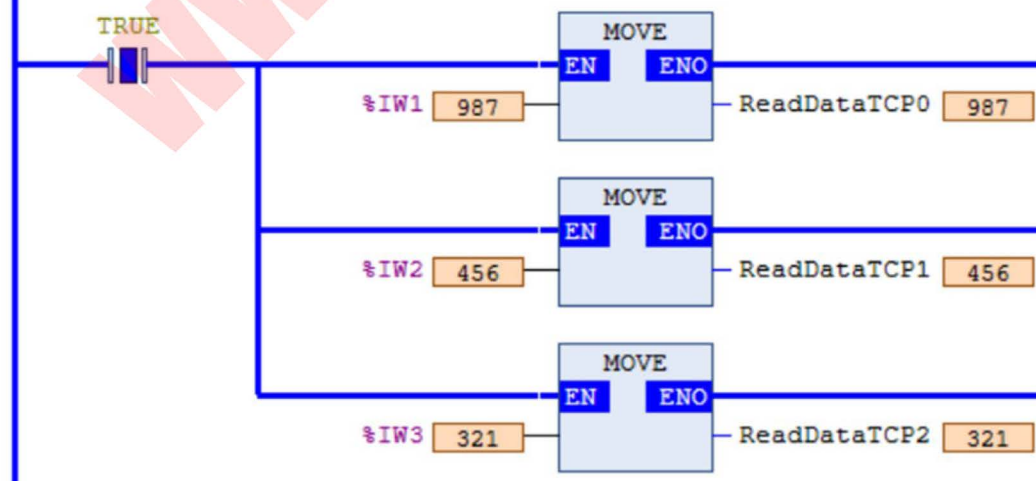
В нашем примере для записи выделены %QW1, %QW2 и %QW3. Для чтения выделены регистры %IW1, %IW2 и %IW3. Данные регистры можно напрямую использовать в программе для обмена данными с Ведомым:

Обмен по Modbus TCP через табличную форму, которая заполняется в софте. Называется Channel.

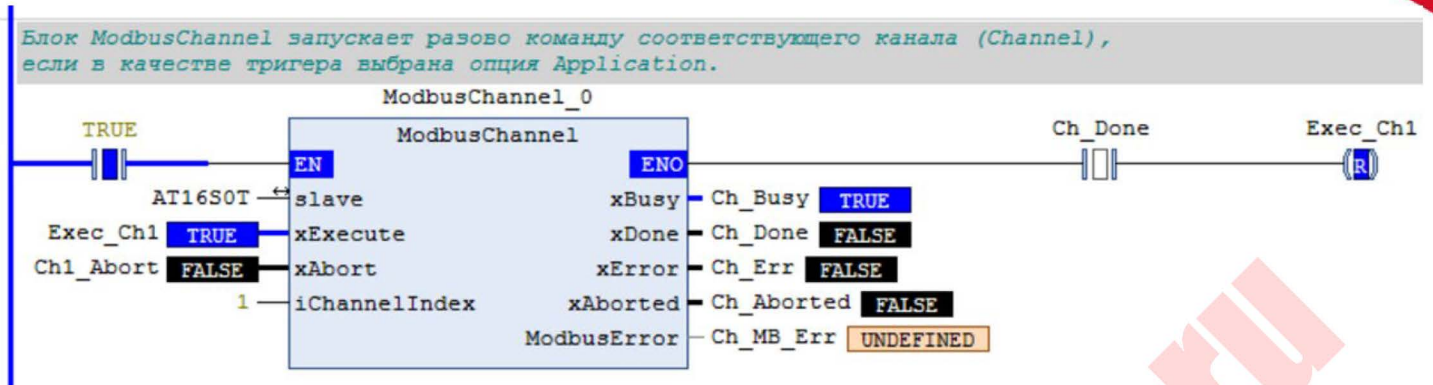
Отправка данных в регистры V3-V5 контроллера Optimus Drive AT16S0T-RU.



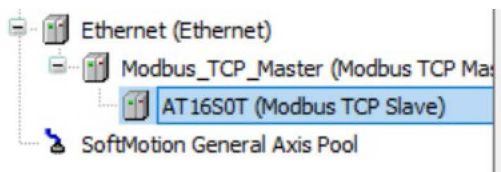
Принятые данные в регистры V3-V5 от контроллера Optimus Drive AT16S0T-RU.



В нашем примере чтение данных запускается через команду **ModbusChannel**:



Ножка **Slave** – это имя узла Modbus TCP Slave. В нашем примере это AT16S0T



Ножка **xExecute** запускает по фронту, т.е. разово блок. Для следующего запроса её надо снова включить.

Ножка **iChannelIndex** содержит номер канала опроса. В нашем примере – 1 (Channel\_1)

Для запуска в автоматическую работу можно использовать конструкцию с инструкцией **Blink**:



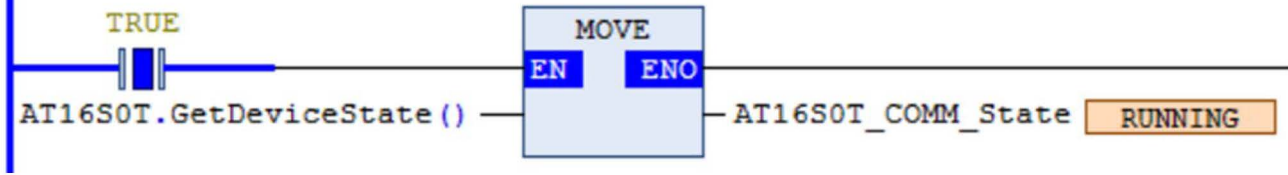
В нашем примере инструкция моргает 50 мс включена и 50 мс выключена. Т.е. общий цикл 100 мс. Выход **Blink** заведён на проверку переднего фронта **R\_TRIG**, которая разово включает ножку **xExecute** блока **ModbusChannel**, который обрабатывает запрос и выставляет флаг на ножке **xDone**, по которой сбрасывается **xExecute**. Таким образом, процесс циклически повторяется пока не будет выключен **Blink**.

Ошибки связи можно проверять через свойство узла **GetDeviceState** и переменную типа **DED.DEVICE\_STATE**

Scope	Name	Address	Data type	Initialization
17	VAR AT16S0T_COMM_State		DED.DEVICE_STATE	

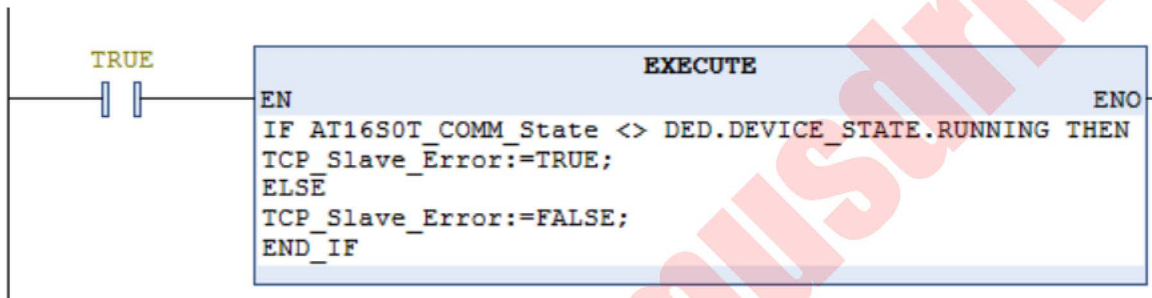


Контроль состояния связи с Ведомым.



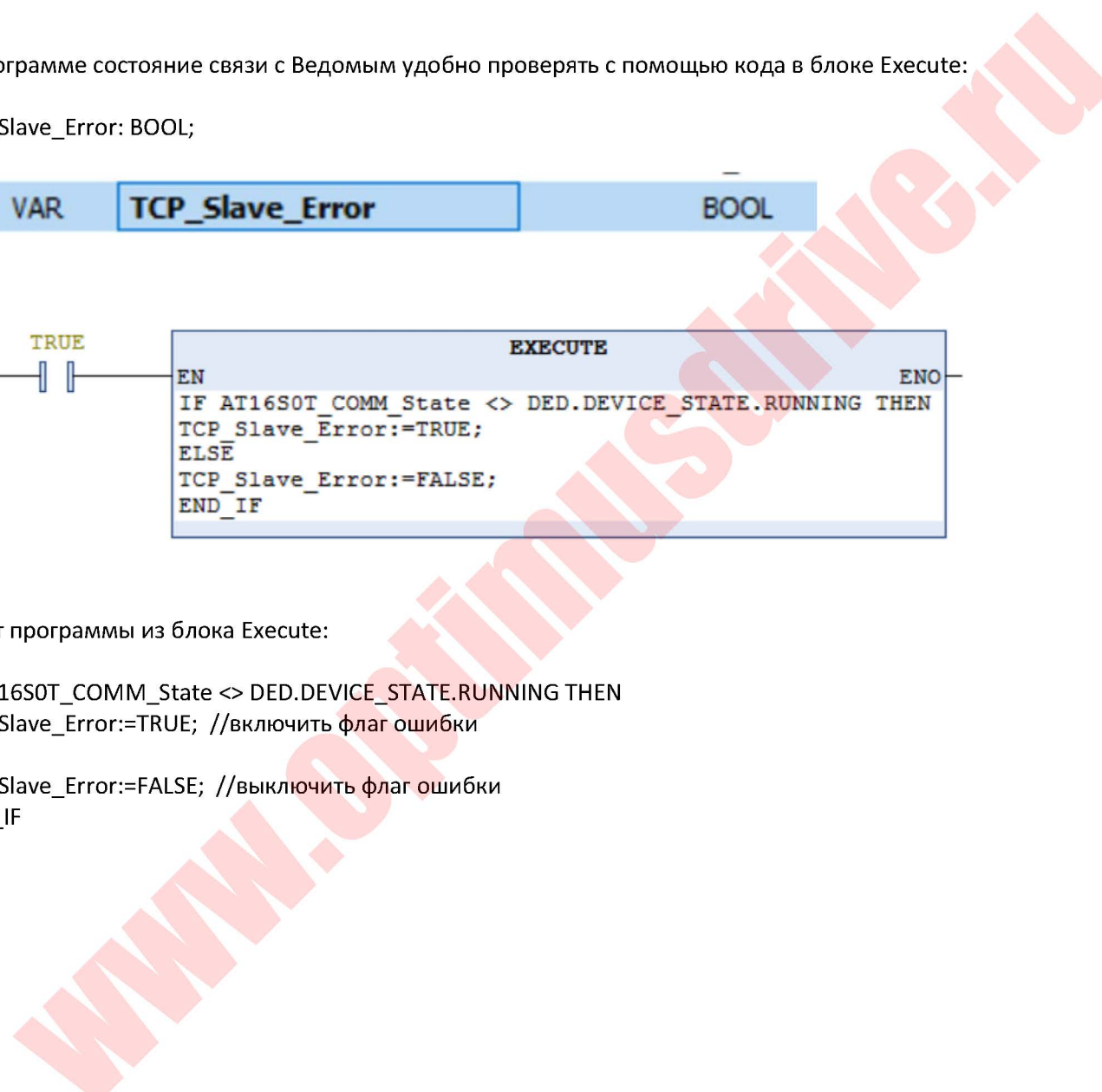
В программе состояние связи с Ведомым удобно проверять с помощью кода в блоке Execute:

TCP\_Slave\_Error: BOOL;



Текст программы из блока Execute:

```
IF AT16S0T_COMM_State <> DED.DEVICE_STATE.RUNNING THEN
TCP_Slave_Error:=TRUE; //включить флаг ошибки
ELSE
TCP_Slave_Error:=FALSE; //выключить флаг ошибки
END_IF
```

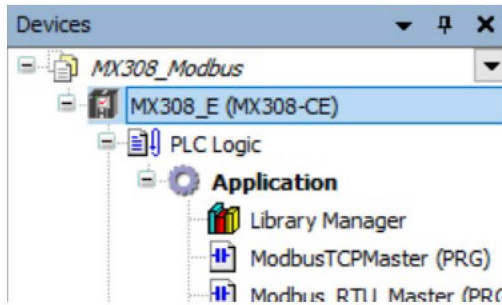




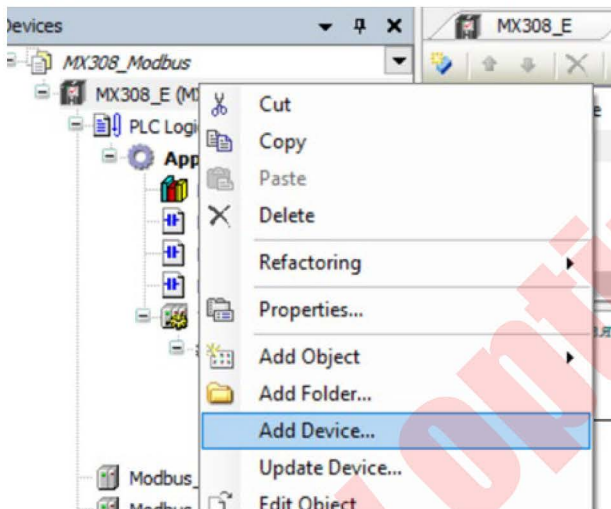
## Связь по протоколу Modbus TCP Slave

Контроллеры серии MX300 могут работать как в режиме Modbus TCP Client (Master), так и в режиме Modbus TCP Server (Slave), причём оба режима могут использоваться одновременно. В данной главе рассматривается организация связи с контроллером по протоколу Modbus TCP Server (Slave).

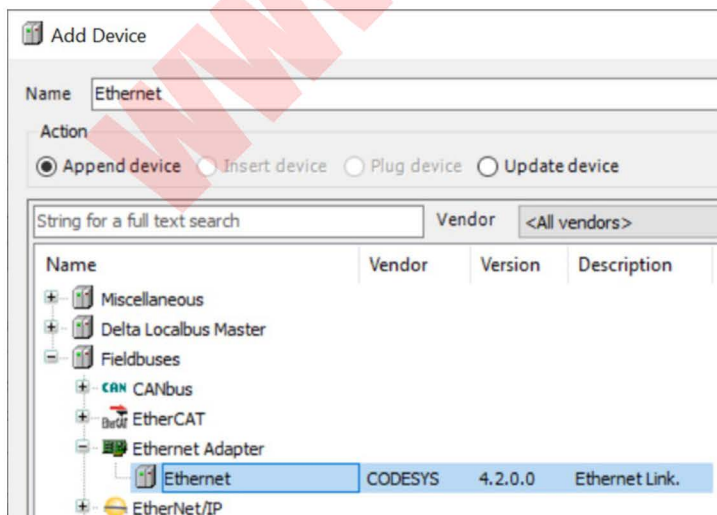
Для начала работы через порт Ethernet необходимо сначала добавить в проект устройство типа **Ethernet**. Для этого встаньте на пункт **Device** и щёлкните правой кнопкой мышки:



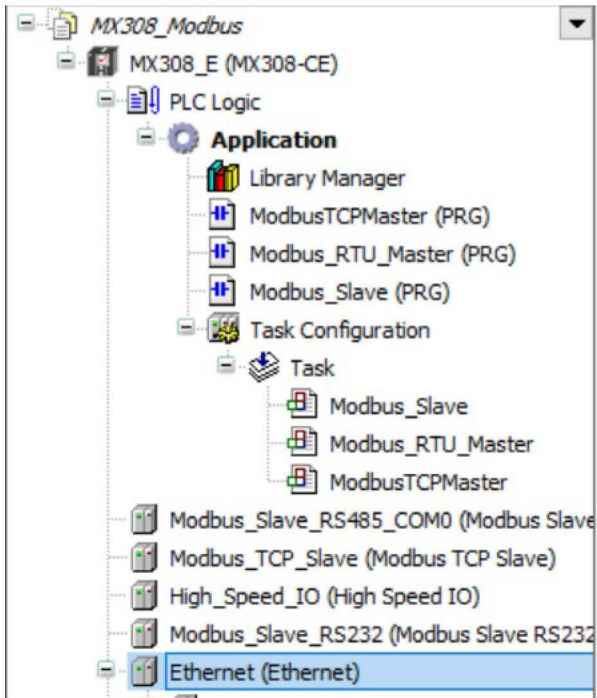
В появившемся меню выберите пункт **Add Device**:



В открывшемся окне выберите пункт **Fieldbuses – Ethernet**:

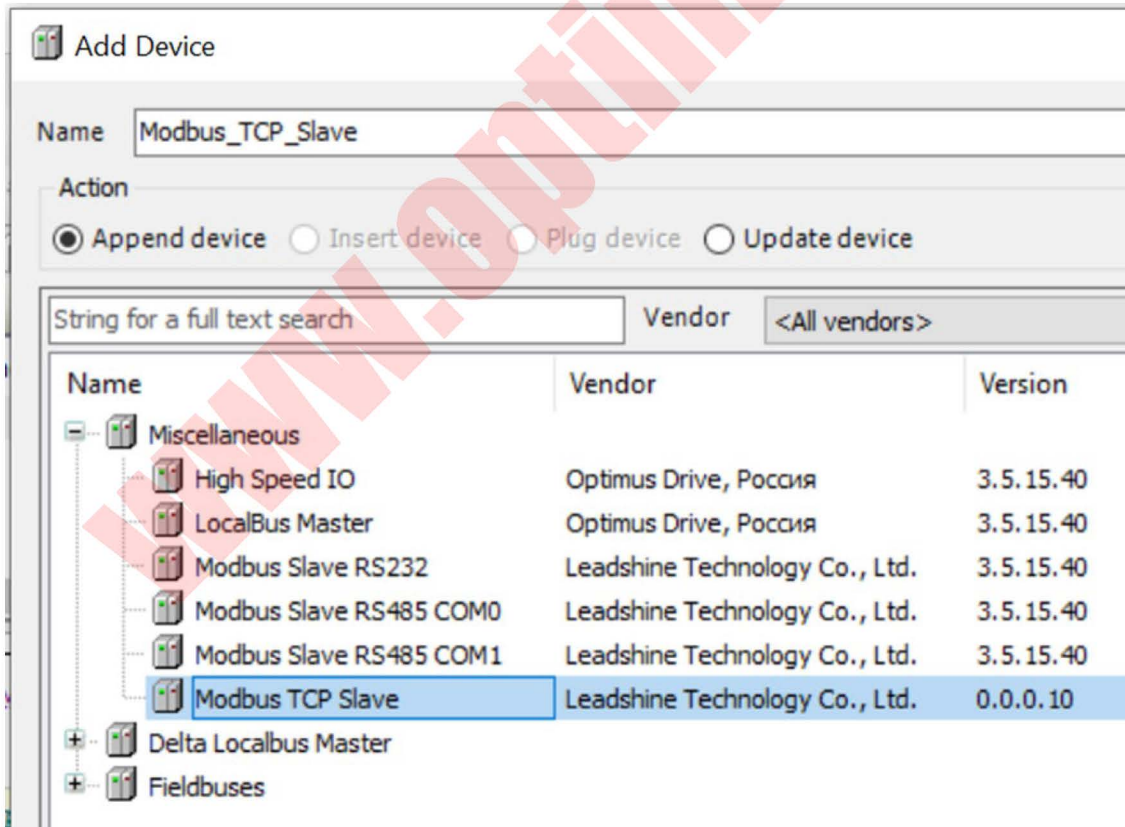


В древе проекта появится пункт **Ethernet**:

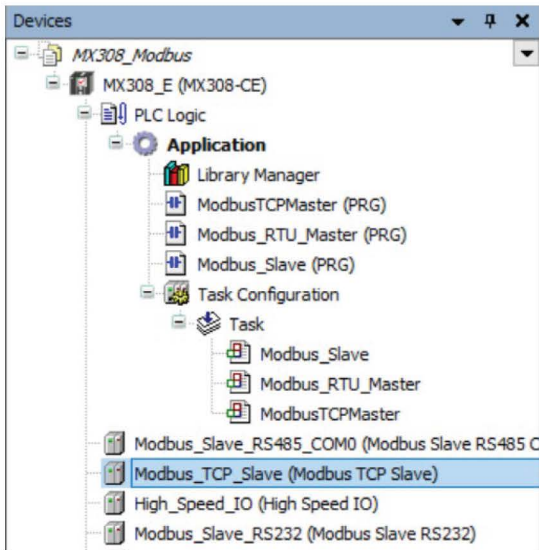


Щёлкните правой кнопкой мышки на пункте Ethernet и в открывшемся меню выберите пункт **Add Device**.

В открывшемся окне выберите пункт **Miscellaneous – Modbus TCP Slave**:



В древе проекта появится пункт Modbus TCP Slave:



После добавления адаптера в проект и его загрузки, контроллер откроет доступ к своей памяти по протоколу Modbus TCP через порт Ethernet. Согласно спецификации протокола Modbus, он имеет возможность читать булевы и словные регистры без обозначения типов данных в них. Поэтому таблица адресов Modbus будет выглядеть следующим образом:

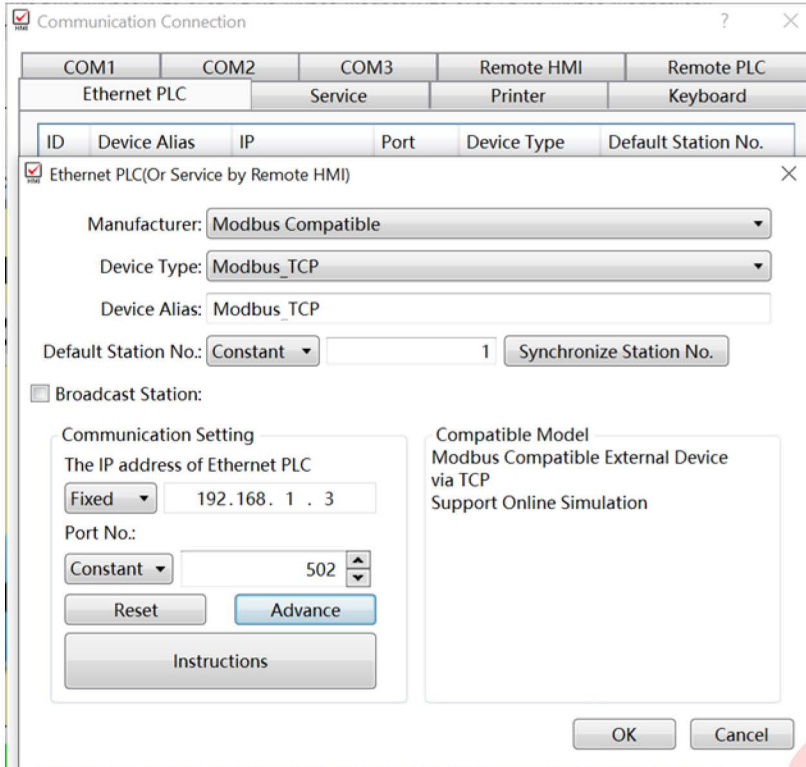
Тип	Диапазон	Функция	Инициализация	Количество
Q (выходы)	(QX0.0 ~ QX8191.7)	0x01,0x05,0x0f	0	65536
I (входы)	(IX0.0 ~ IX8191.7)	0x02	0	65536
M (данные)	%MW0~%MW65535	0x03,0x06,0x10	0	65536

Регистры типа %MW0~%MW65535 являются стандартными словами 16 бит. Адресация к битам в словах не поддерживается. Поэтому в программе биты типа %MX0.0... можно использовать, но их состояние нужно передавать как словный регистр, а потом на стороне Мастера разбирать по битам.

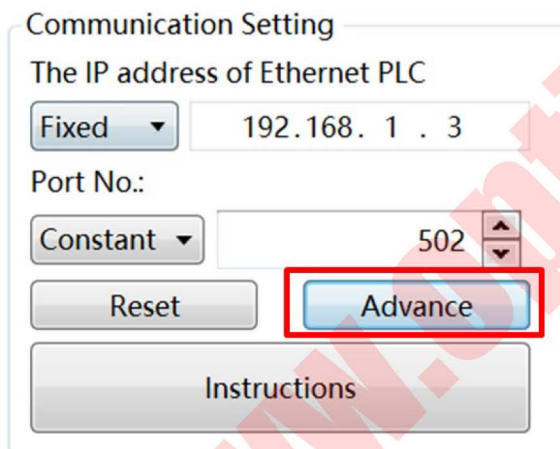
Например, маркеры %MX0.0.. %MX0.7 и %MX1.0.. %MX1.7 будут входить в состав регистра %MW0, маркеры %MX2.0.. %MX2.7 и %MX3.0.. %MX3.7 будут входить в состав регистра %MW1 и т.д.

Контроллеры серии MX300 могут выступать в качестве Ведомого устройства для любого стандартного Мастера Modbus TCP. Например, можно рассмотреть в связь с панелью оператора Optimus Drive VI20-070S-FE-RU.

В панели необходимо выбрать стандартный Modbus TCP драйвер:



Указать IP адрес контроллера. В нашем примере 192.168.1.3 и во вкладке **Advance** убрать смещение адреса:

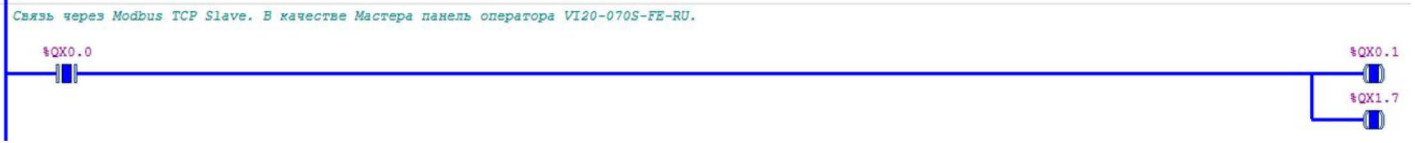


Max Bit Registers: 64

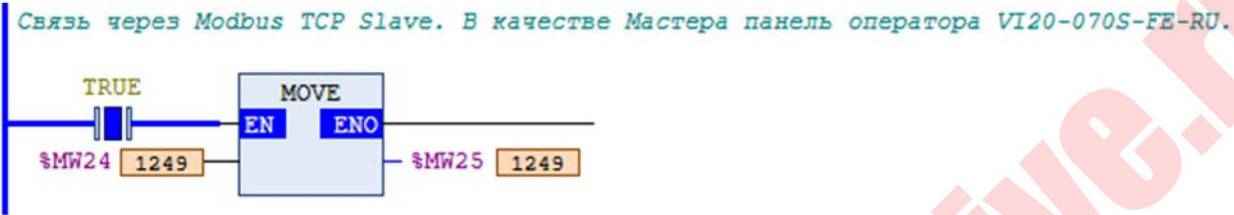
Time Interval: 5

Base Address: 0

В программе контроллера создана простая программа для отображения данных, в которой задействованы следующие регистры:

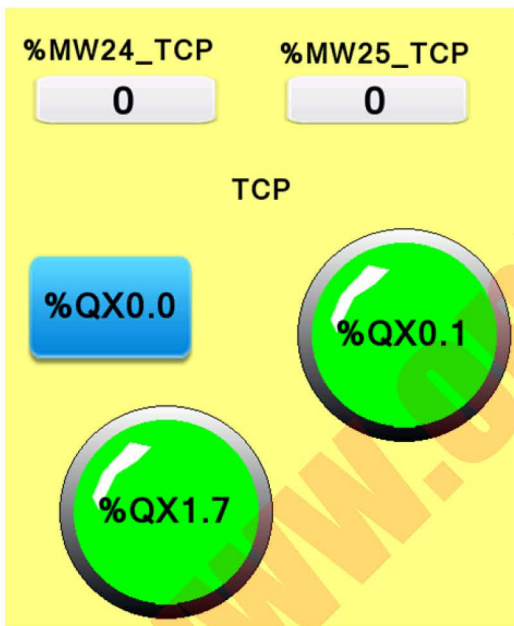


Регистр %QX0.0 - контакт, регистры %QX0.1 и %QX1.7 - выходные катушки.

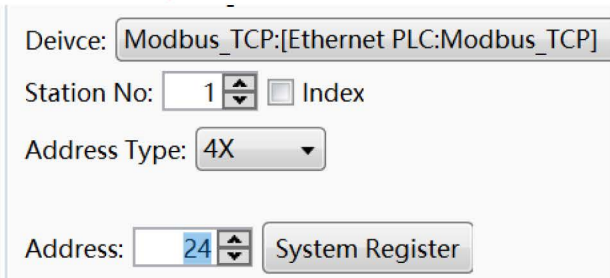


Регистр %MW24 - источник данных, регистр %MW25 - приёмник данных.

В панели оператора нарисован простой экран с пятью объектами соответственно: Input, Display, Button и 2 штуки Indicator.



В панелях используется десятичное задание адреса, поэтому обращение к регистру %MW24 будет выглядеть так:



а к регистру %MW25 так

Device: Modbus\_TCP:[Ethernet PLC:Modbus\_TCP]  
 Station No: 1  Index  
 Address Type: 4X  
 Address: 25 System Register

Таким образом, нужно просто указать номер регистра %MW\*\*\*\*.

Обращение к булеву регистру %QX0.0 выглядит так:

Device: Modbus\_TCP:[Ethernet PLC:Modbus\_TCP]  
 Station No: 1  Index  
 Bit-index within a Byte Register  
 Address Type: 0X  
 Address: 0 Sys

к регистру %QX0.1 так:

Standard Bit Address Input  
 Use Address Tag  
 Device: Modbus\_TCP:[Ethernet PLC:Modbus\_TCP]  
 Station No: 1  Index  
 Bit-index within a Byte Register  
 Address Type: 0X  
 Address: 1 System Register  
 Format(Range):DDDDD(0~65534)

а к регистру %QX1.7 так:

Standard Bit Address Input  
 Use Address Tag  
 Device: Modbus\_TCP:[Ethernet PLC:Modbus\_TCP]  
 Station No: 1  Index  
 Bit-index within a Byte Register  
 Address Type: 0X  
 Address: 15 System Register

Т.е. идёт сплошная десятичная адресация. Каждый последующий регистр увеличивает адрес на 1.

%QX0.0.. %QX0.7 имеют десятичные адреса 0..7  
%QX1.0.. %QX1.7 имеют десятичные адреса 8..15  
%QX2.0.. %QX2.7 имеют десятичные адреса 16..23  
и т.д.

При начале опроса контроллера панель будет отображать состояние регистров:

%MW24\_TCP

1249

%MW25\_TCP

1249

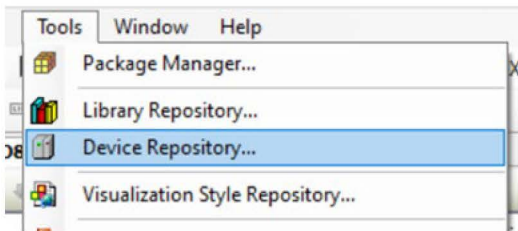
TCP



## Работа с преобразователем частоты серии Optimus Drive AD800 по сети EtherCAT

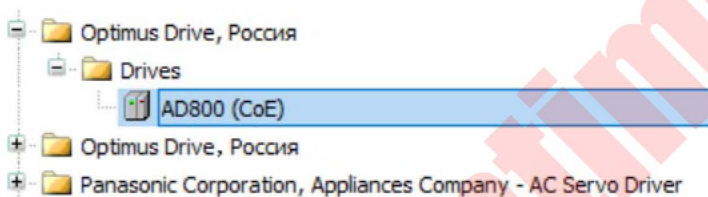
Преобразователи частоты (ПЧ) серии **AD800** имеют возможность расширяться функциональными платами. Для интеграции ПЧ в сеть EtherCAT используется плата **ET01**. Установка и работа с данной платой приведены в Руководстве по эксплуатации «**Плата расширения EtherCAT ET01**».

Для включения ПЧ в состав проекта контроллера серии MX300 необходимо наличие в проекте файла с описанием устройства, или как его ещё называют XML файла EtherCAT. Данный файл входит в состав пэкиджа для контроллера, но при необходимости загрузки другой версии файла нужно воспользоваться меню **Tools – Device Repository**:



Нажать кнопку **Install**, выбрать нужный файл и установить в проект.

Файл с описанием будет лежать по пути **Fieldbuses – EtherCAT – Slave – Optimus Drive**:



Параметры ПЧ AD800 для работы по EtherCAT:

P7-00 - Сброс на заводские установки (передёрнуть питание).

P0-10 = 3,

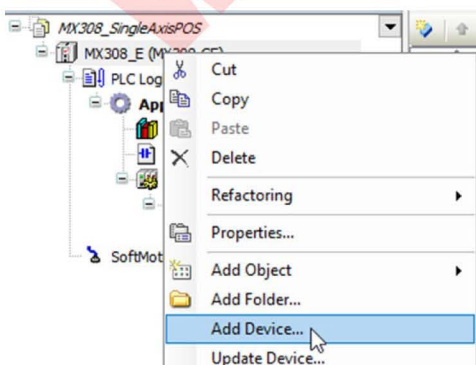
P0-11 = 20,

P0-17 = 0,

P0-18 = 2,

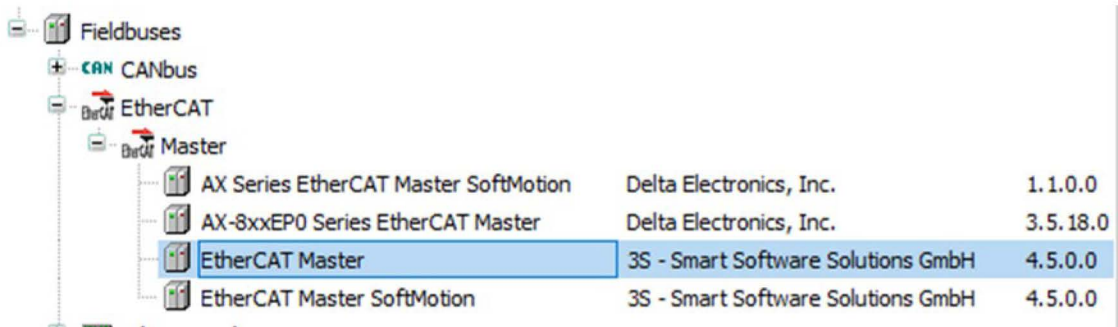
P0-29 = 1

Далее необходимо добавить в проект EtherCAT адаптер. Для этого щёлкните правой кнопкой на названии контроллера и в меню выберите пункт **Add Device**:

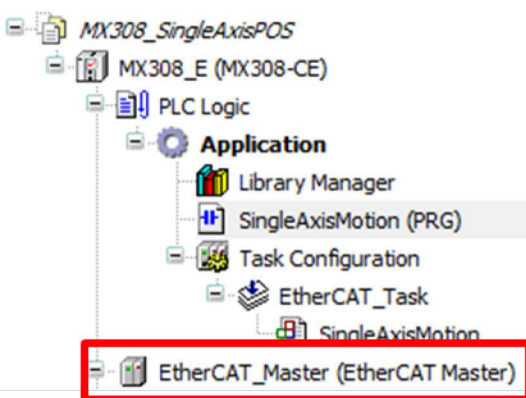




В открывшемся окне выберете пункт **EtherCAT - Master - EtherCAT Master**:

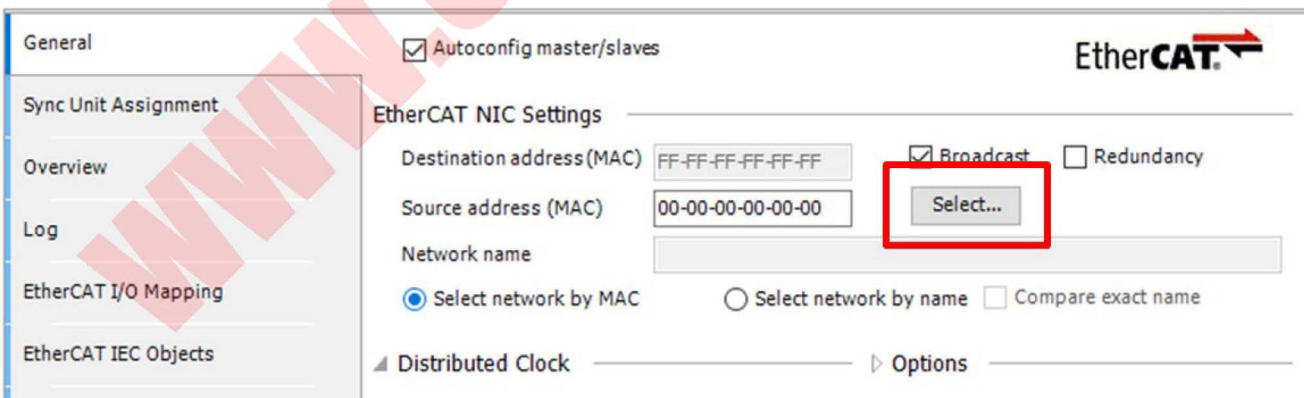


В древе проекта появится пункт:

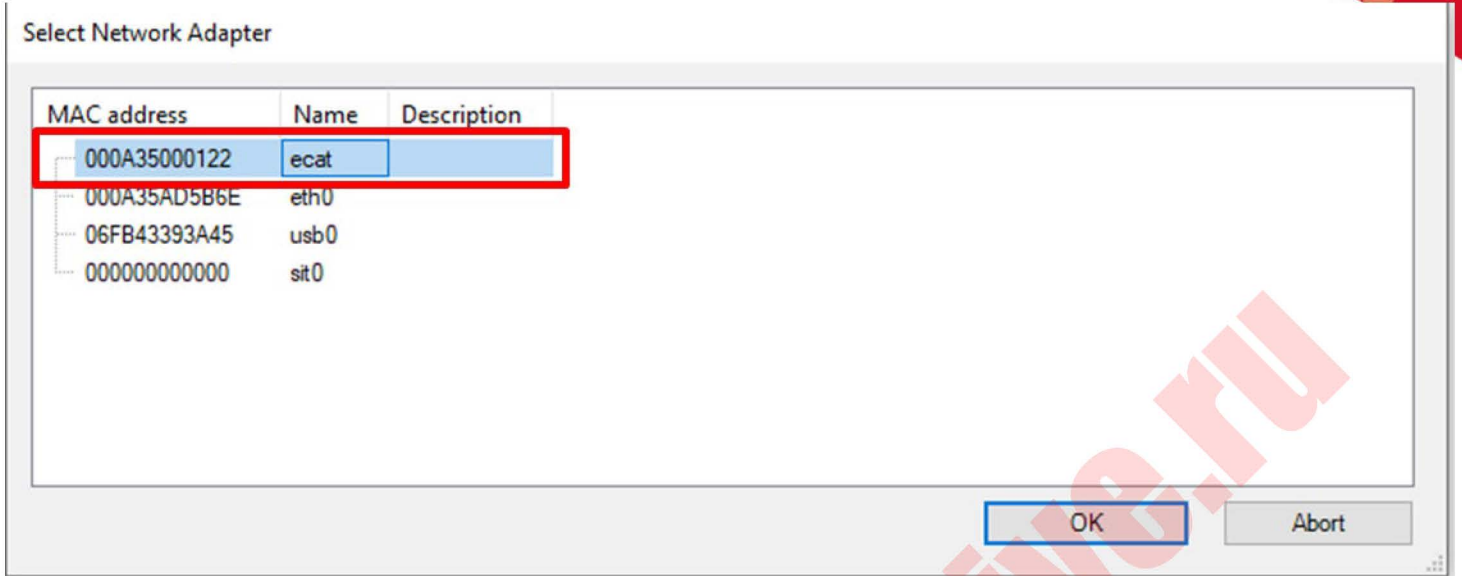


Далее необходимо назначить в проекте Мастера сети EtherCAT. В нашем примере это будет контроллер MX300, поэтому необходимо установить с ним связь (см. соответствующий раздел данного Руководства):

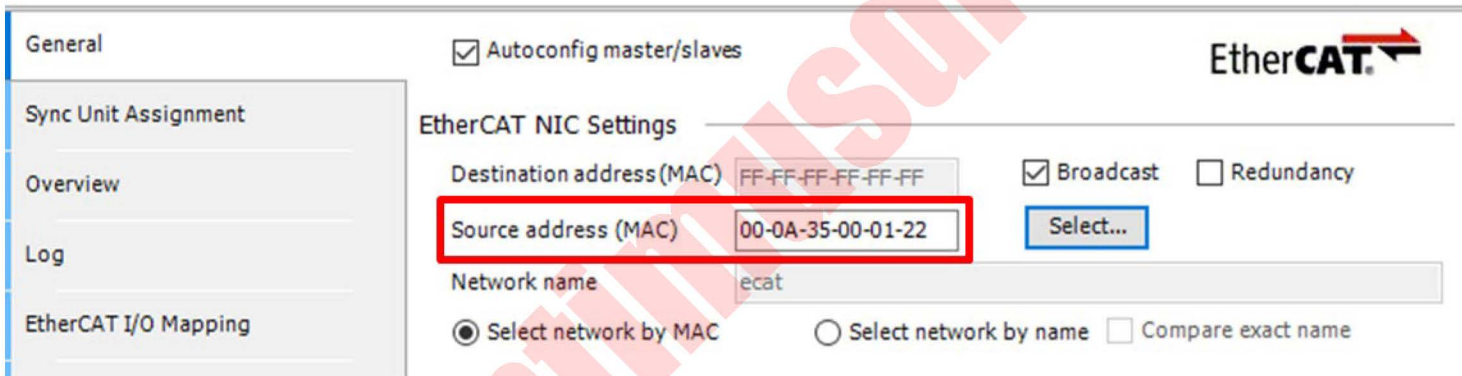
Далее щёлкните дважды левой кнопкой мышки на пункте **EtherCAT Master** и в открывшейся вкладке выберите пункт **General** и нажмите кнопку **Select**:



В открывшемся окне выберите пункт (MAC адрес у каждого контроллера свой):

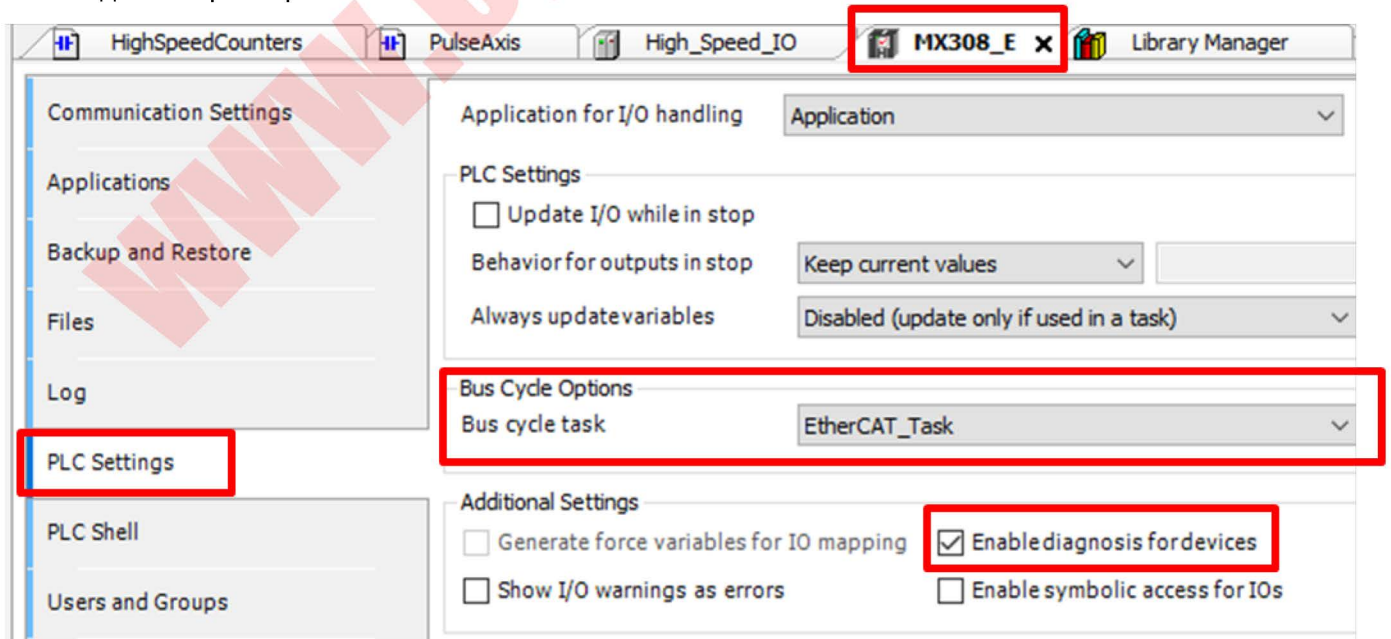


MAC адрес должен установиться в данном поле:

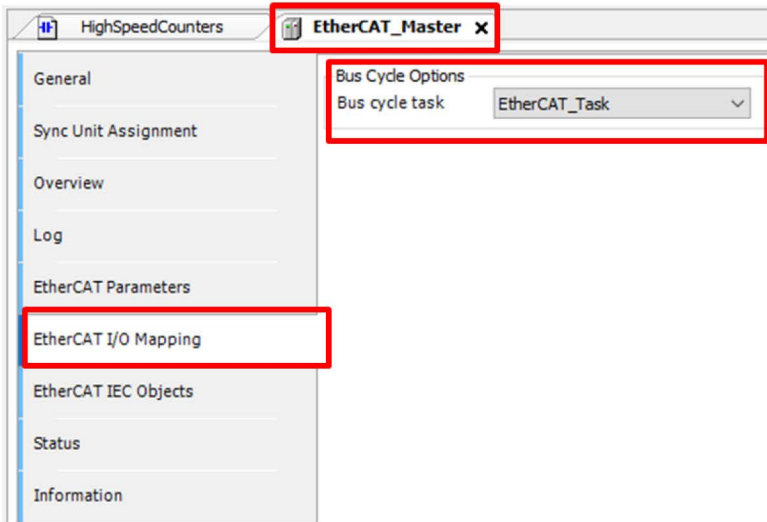


Далее необходимо активировать задачу EtherCAT\_Task. Для этого сделайте следующие настройки:

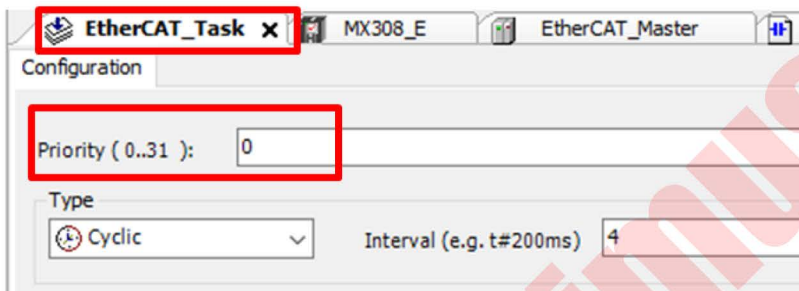
Во вкладке Контроллера:



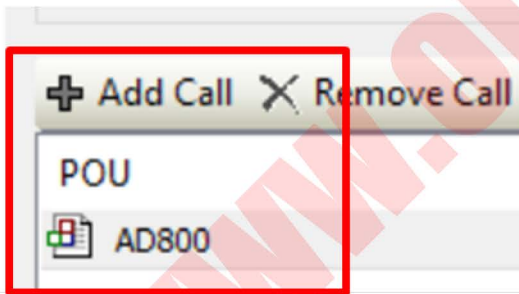
Во вкладке EtherCAT\_Master:



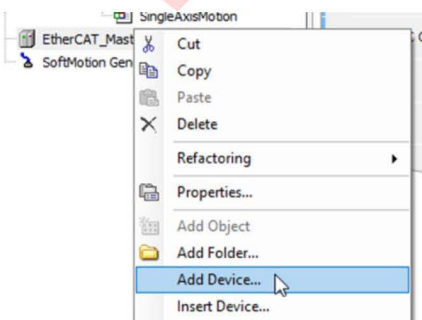
Во вкладке задачи выставить высший приоритет (0), привязать нужные POU:



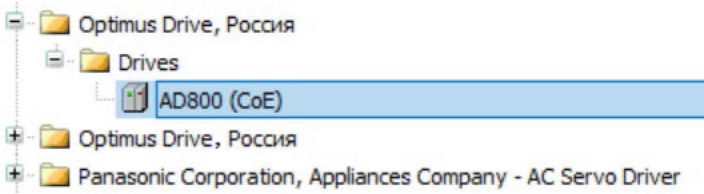
*Для приводов AD800 рекомендуется такт шины не менее 4 мс. При большом количестве приводов на шине время опроса необходимо увеличить.*



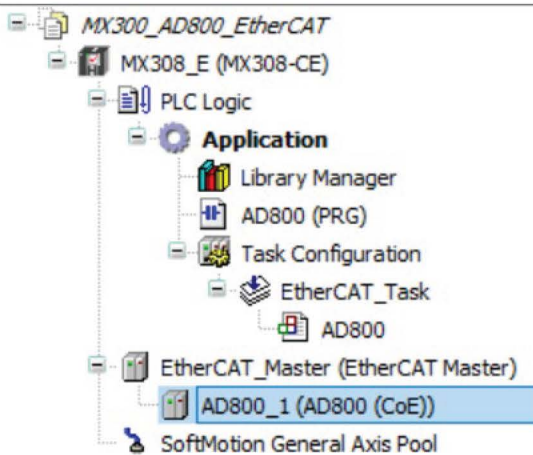
Далее необходимо добавить в проект сервопривод. Для этого щёлкните правой кнопкой мышки на пункте **EtherCAT Master** и в открывшемся окне выберите пункт **Add Device**:



В открывшемся окне выберите Преобразователь Частоты Optimus Drive AD800: по пути **Fieldbuses – EtherCAT – Slave – Optimus Drive**:



В древе проекта появится пункт AD800:



На данном этапе мы добавили привод как узел сети EtherCAT и после загрузки проекта в контроллер привод станет доступен для управления.

Преобразователи частоты добавляются в проект как обычные устройства, т.е. управляются напрямую через PDO пакеты. Параметрам ПЧ будут поставлены в соответствие регистры контроллера, записывая нужные значения в которые, можно будет управлять приводом.

Щёлкните дважды левой кнопкой мышки на названии узла (AD800) и в открывшейся вкладке выберите пункт **Process Data**:

General	Select the Outputs			Select the Inputs		
Process Data	<input checked="" type="checkbox"/> 16#1610 Outputs			<input checked="" type="checkbox"/> 16#1A10 Inputs		
Log	ControlWord	UINT	16#2001:16#00	Status_Word_P9-01	UINT	16#3385:16#00
	Set_Value	UINT	16#2002:16#00	Output_Frequency_P9-07	UINT	16#338B:16#00

По умолчанию назначаются два регистра на отправку и два на приём данных от привода.

В разделе **Outputs**, т.е. данные в сторону привода, будет два регистра – **ControlWord** (команды приводу) и **Set\_Value** (задание скорости в Герцах).

В разделе **Inputs**, т.е. данные от привода в сторону контроллера, будет также два регистра – **Status\_Word** (слово состояния привода) и **Output\_Frequency** (выходная частота в Герцах).

При необходимости, в PDO пакет можно добавить ещё параметры. Для этого необходимо включить экспертный режим:

Далее необходимо щёлкнуть мышкой на пункте **Expert Process Data** и выбрать нужную группу, например **Inputs**, и нажать кнопку **Insert**. В открывшемся окне нужно ввести нужный адрес регистра, который Вы хотите добавить. Адрес привязан к параметрам ПЧ и вычисляется путём сложения 0x3000 + номер параметра в HEX. Мониторинговые параметры находятся в группе 9. Например, выходной ток находится в параметр P9-08. Переводим число 908 в HEX, получаем 0x38C, складываем с 0x3000 и получаем число 0x338C и вводим в форму:

Для данного параметра тип данных нужно установить UNIT. Адрес (Index) и название (Name) вводятся строго в латинской раскладке клавиатуры.

Ещё для примера добавим напряжение на шине DC. Данный параметр находится в P9-11. Переводим число 911 в HEX, получаем 0x38F, складываем с 0x3000 и получаем число 0x338F, нажимаем **Insert** и в открывшемся окне вводим в формате UINT:

В итоге будет список параметров:

Index	Size	Offs	Name
16#3385:16#00	2.0	0.0	Status_Word_P9-01
16#338B:16#00	2.0	2.0	Output_Frequency_P9-07
16#338C:16#00	2.0	4.0	Output_Current_P9-08
16#338F:16#00	2.0	6.0	DC_BUS_Voltage_P9-11
		8.0	

Далее в пункте EtherCAT I/O Mapping можно посмотреть регистры контроллера, через которые будет осуществляться управление приводом:

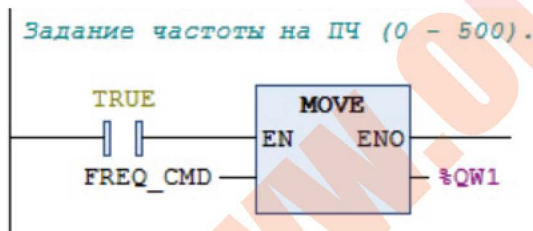
General	Find	Filter	Show all	
Variable	Mapping	Channel	Address	Type
16#1610 Outputs				
		ControlWord	%QW0	UINT
		Set_Value	%QW1	UINT
16#1A10 Inputs				
		Status_Word_P9-01	%IW0	UINT
		Output_Frequency_P9-07	%IW1	UINT
		Output_Current_P9-08	%IW2	UINT
		DC_BUS_Voltage_P9-11	%IW3	UINT

Поверх них можно назначить теги или использовать в программе напрямую.

А можно объявить в программе переменные и через них работать с регистрами, отвечающими за привод:

Scope	Name	Address	Data type	Initialization
VAR	CONTROL_WORD		UINT	
VAR	FREQ_CMD		UINT	
VAR	OUTPUT_FREQ		UINT	
VAR	OUTPUT_CURRENT		UINT	
VAR	DC_BUS_VOL		UINT	
VAR	STATUS_WORD		UINT	
VAR	RUN_STOP		BOOL	
VAR	FWD_REV		BOOL	

И отправлять данные в соответствующий регистр, например задание частоты:



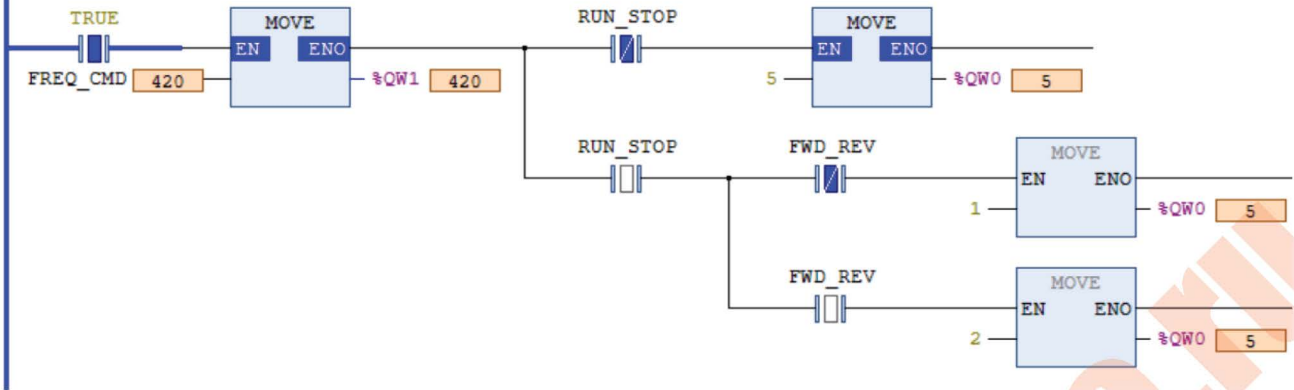
Для управления приводом в контрольное слово нужно записать следующие значения:

- 5 – Стоп
- 1 – Вращение вперед
- 2 – Вращение назад

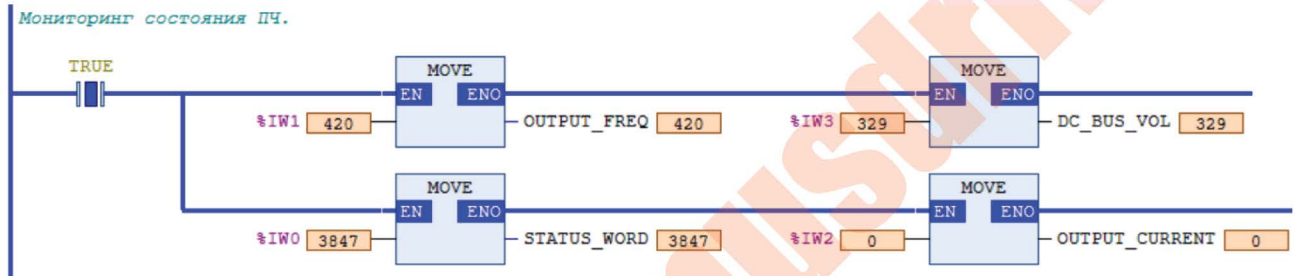
(другие команды можно посмотреть в Руководстве по эксплуатации платы ET01).

Для удобства работы команды можно описать в программе через булевы переменные, которые будут отправлять нужное число в командный регистр, например такой код:

Задание частоты на ПЧ (0 - 500). Команда RUN - STOP (Работа RUN\_STOP = TRUE) и выбор вперед-назад. Назад FWD\_REV = TRUE.



Мониторинг состояния ПЧ можно осуществлять также через переменные:



Помимо циклического опроса через пакет типа PDO, привод позволяет опрашивать себя посредством запросов типа SDO (запрос-ответ). Это полезно для конфигурирования привода, которое делается один раз и далее не должно занимать трафик. С этой целью используются команды **ETC\_CO\_SdoWrite** и **ETC\_CO\_SdoRead**.

Для примера запишем в привод и для контроля прочитаем 1-е время разгона, это параметр P0-51. Адрес для EtherCAT вычисляется также как для пакетов PDO (см. выше по тексту).

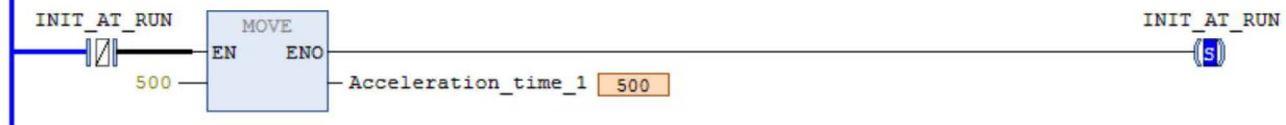
$0x3000 + 0x0033 = 0x3033$  это EtherCAT адрес параметра P0-51

Блок записи значения времени разгона в привод:

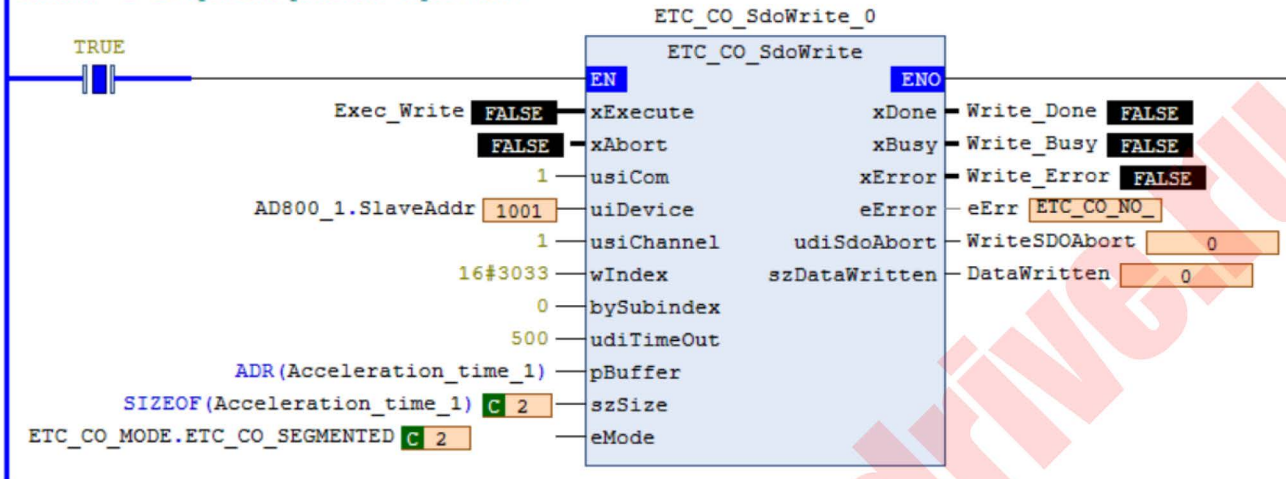
Scope	Name	Address	Data type	Initialization
VAR	INIT_AT_RUN		BOOL	
VAR	ETC_CO_SdoWrite_0		ETC_CO_SdoWrite	
VAR	Acceleration_time_1		UINT	
VAR	Exec_Write		BOOL	
VAR	Write_Done		BOOL	
VAR	Write_Busy		BOOL	
VAR	Write_Error		BOOL	
VAR	eErr		ETC_CO_ERROR	
VAR	WriteSDOAbort		UDINT	
VAR	DataWritten		UDINT	



Инициализация значения по умолчанию в переменную 1-го времени ускорения Acceleration\_time\_1.



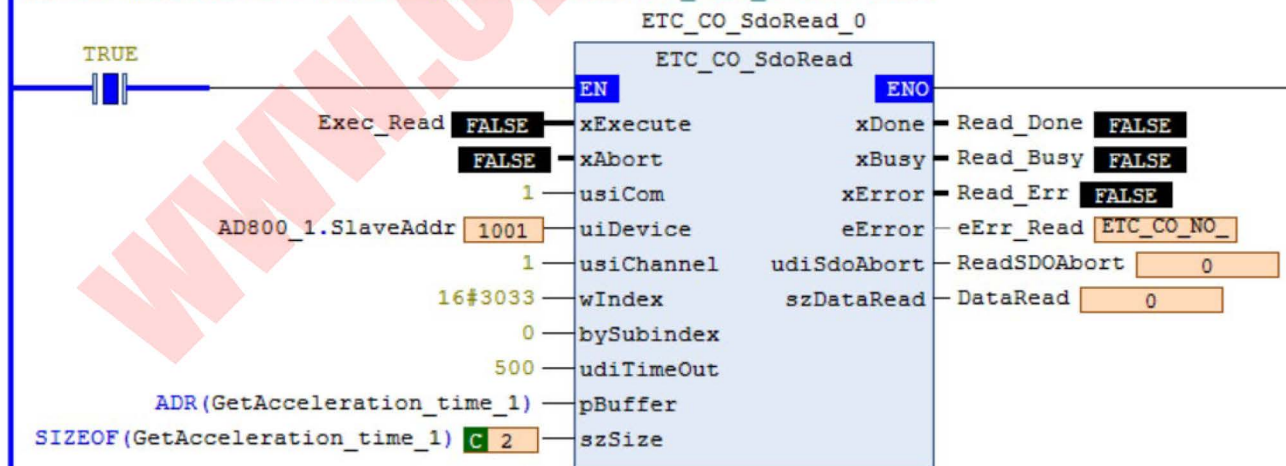
Запись 1-го времени разгона через SDO.



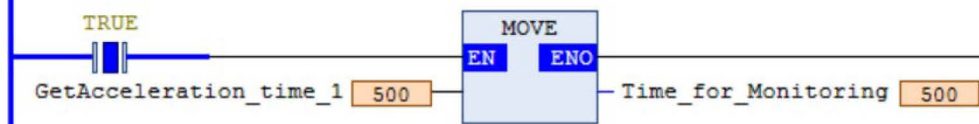
Блок чтения значения времени разгона из привода:

- ◆ VAR ETC\_CO\_SdoRead\_0 ETC\_CO\_SdoRead
- ◆ VAR Exec\_Read BOOL
- ◆ VAR Read\_Done BOOL
- ◆ VAR Read\_Busy BOOL
- ◆ VAR Read\_Err BOOL
- ◆ VAR eErr\_Read ETC\_CO\_ERROR
- ◆ VAR ReadSDOAbort UDINT
- ◆ VAR DataRead UDINT
- ◆ VAR GetAcceleration\_time\_1 UINT
- ◆ VAR Time\_for\_Monitoring UINT

Чтение заданного 1-го времени разгона через SDO для проверки записи на предыдущем шаге. Данные принимаются в переменную GetAcceleration\_time\_1 типа UINT.



Для мониторинга.





## Связь по протоколу Ethernet/IP Scanner (Master)

Протокол **Ethernet/IP** (IP = Industrial Protocol) является одним из наиболее распространённых промышленных протоколов в мире и поддерживается сотнями производителей промышленного оборудования. **Ethernet/IP** (сокращённо **EIP**) использует модель **CIP** (Common Industrial Protocol) поверх стандартного интерфейса **Ethernet** и работает по схеме **Producer/Consumer**. На транспортном уровне используется протокол **UDP**.

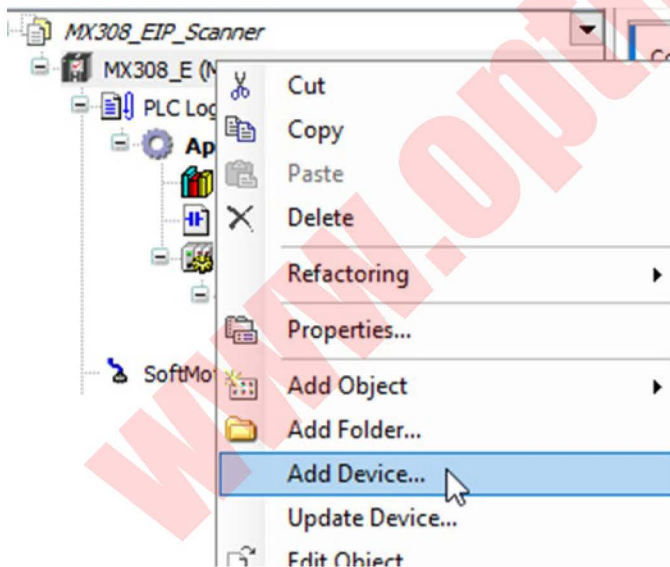
Контроллеры серии MX300 могут работать как в режиме **Ethernet/IP Scanner (Master)**, так и в режиме **Ethernet/IP Adapter (Slave)**. В данной главе рассматривается организация связи контроллера с Ведомыми устройствами в режиме Ethernet/IP Scanner (Master).

Для использования Ethernet/IP к проекту должны быть подключены следующие библиотеки: IoDrvEthernet, IoDrvEtherNetIP, EtherNetIP Services и DED.

CAA Device Diagnosis = CAA Device Diagnosis, 3.5.15.0 (CAA Technical Workgroup)	DED	3.5.15.0
EtherNetIP Services = EtherNetIP Services, 4.5.0.0 (3S - Smart Software Solutions GmbH)	ENIP	4.5.0.0
IoDrvEthernet = IoDrvEthernet, 4.2.0.0 (3S - Smart Software Solutions GmbH)	IoDrvEthernet	4.2.0.0
IoDrvEtherNetIP = IoDrvEtherNetIP, 4.5.1.0 (3S - Smart Software Solutions GmbH)	IoDrvEtherNetIP	4.5.1.0

Настройка связи по протоколу Ethernet/IP начинается с добавления к проекту адаптера Ethernet и привязка его к определённому порту.

Щёлкните в древе правой кнопкой мышки на пункте **Device (MX308\_E)** и в отрывшемся меню выберите пункт **Add Device**



В открывшемся окне выберите **Fieldbuses – Ethernet**:

Add Device

Name Ethernet\_1

Action

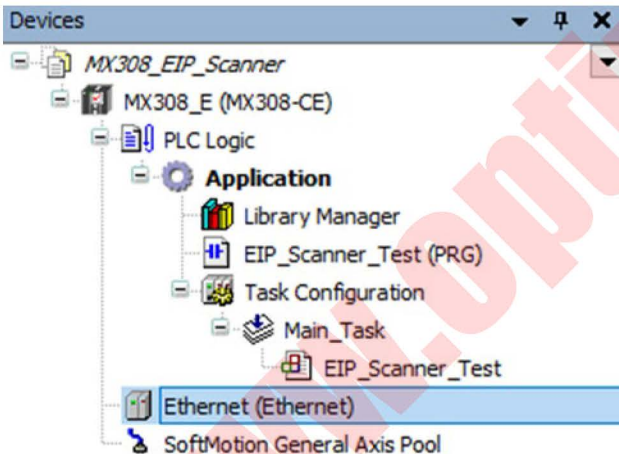
Append device  Insert device  Plug device  Update device

String for a full text search

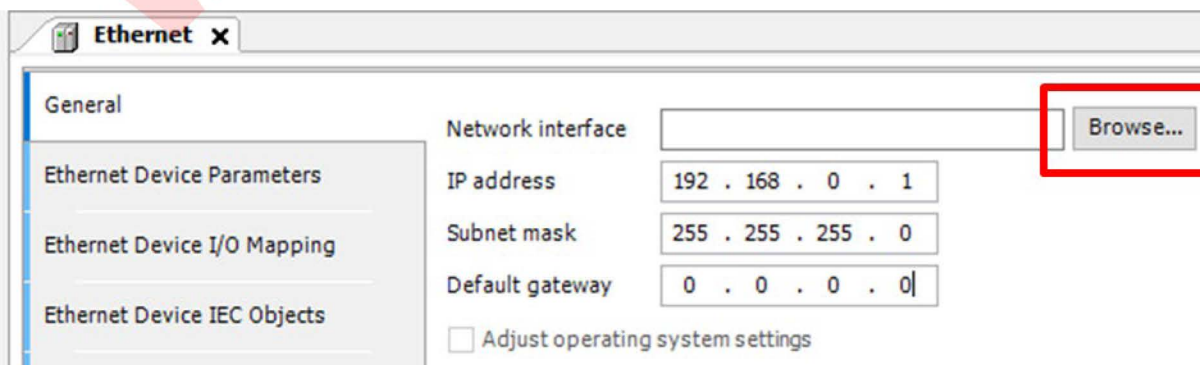
Vendor <All vendors>

Name	Vendor	Version	Description
Miscellaneous			
Delta Localbus Master			
Fieldbuses			
CANbus			
EtherCAT			
Ethernet Adapter			
<b>Ethernet</b>	CODESYS	4.2.0.0	Ethernet Link.
EtherNet/IP			
Home&Building Automation			
Modbus			
Profinet IO			

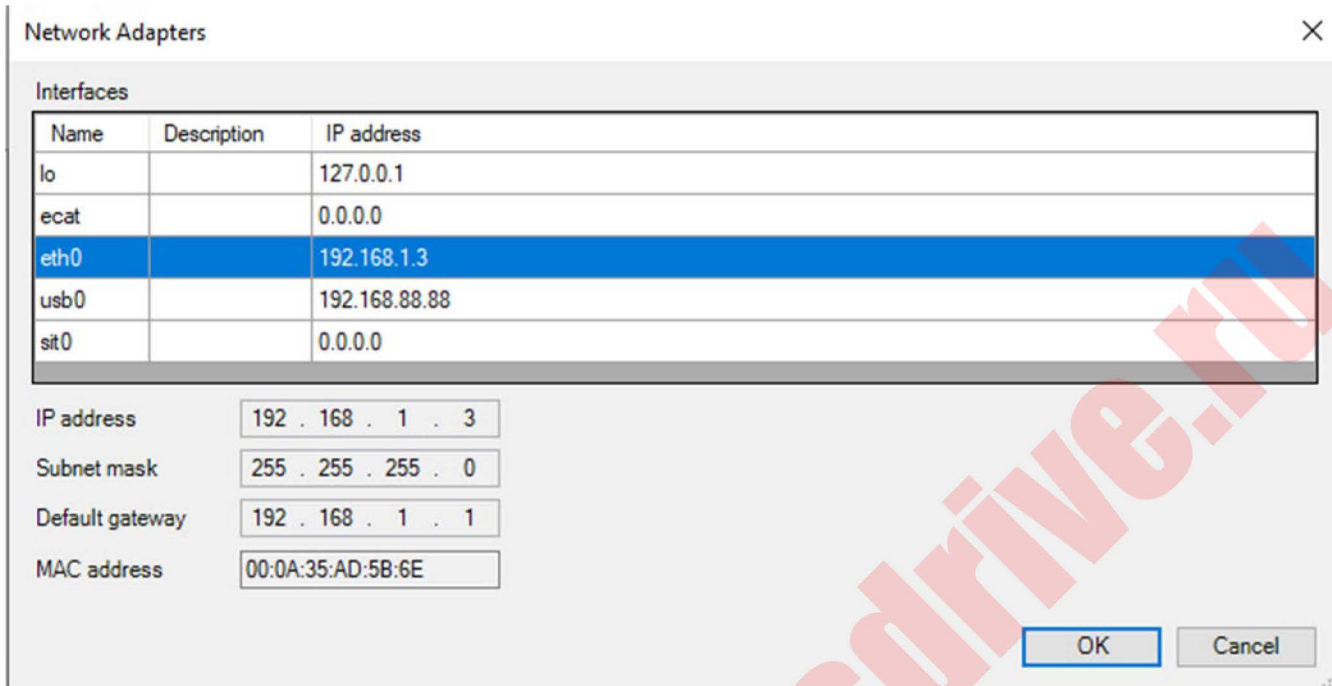
В древе проекта появится пункт **Ethernet**:



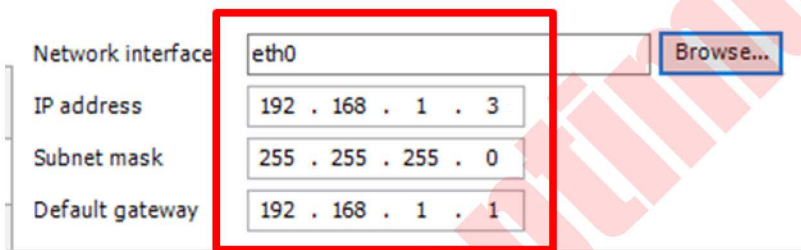
Щёлкните на нём дважды левой кнопкой мышки и в открывшейся вкладке выберите пункт **General**:



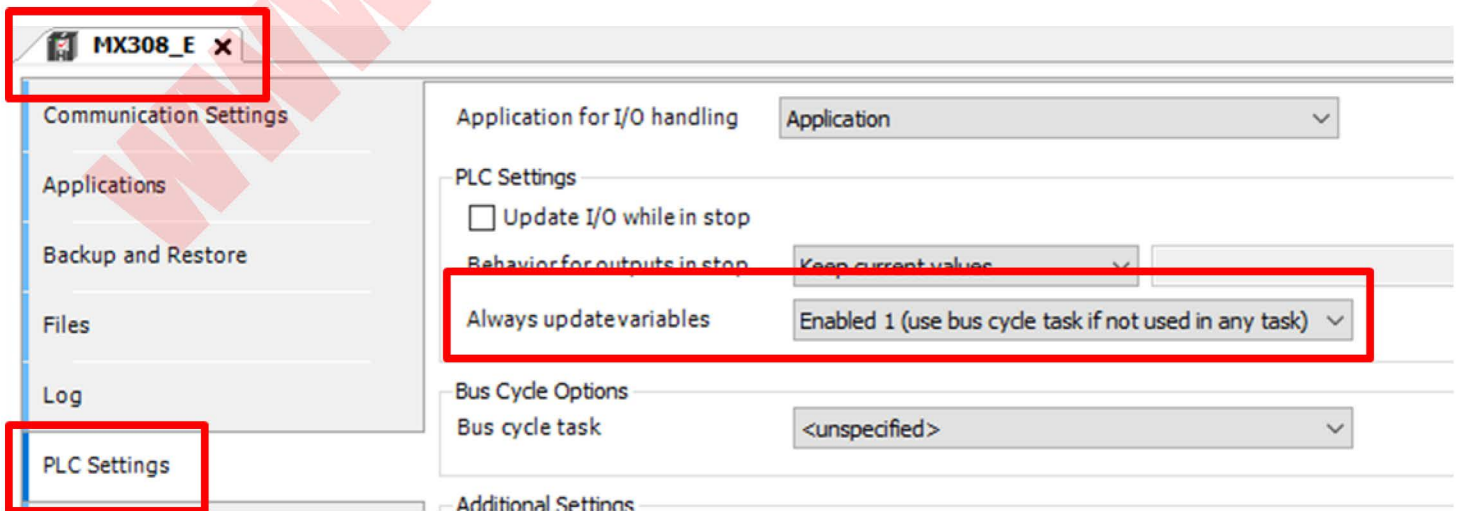
Нажмите кнопку **Browse** и в открывшемся окне выберите IP адрес контроллера:



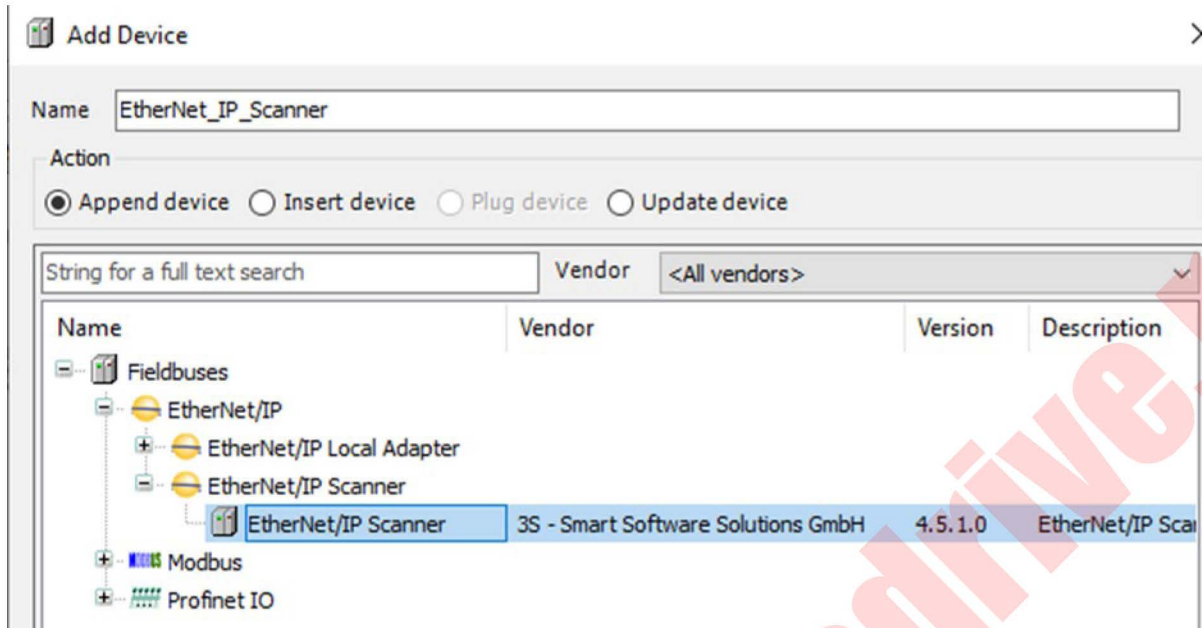
После чего Адаптер будет привязан к Ethernet порту с конкретным IP адресом:



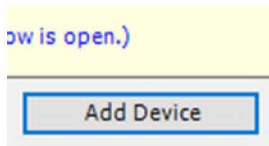
Далее двойным щелчком левой кнопки мыши откройте вкладку **Device** и в пункте **PLC Settings** разрешите обновление переменных. Данный шаг позволяет видеть изменение переменных и регистров без написания в теле программы, т.е. в таблице Watch и в таблицах мэпинга.



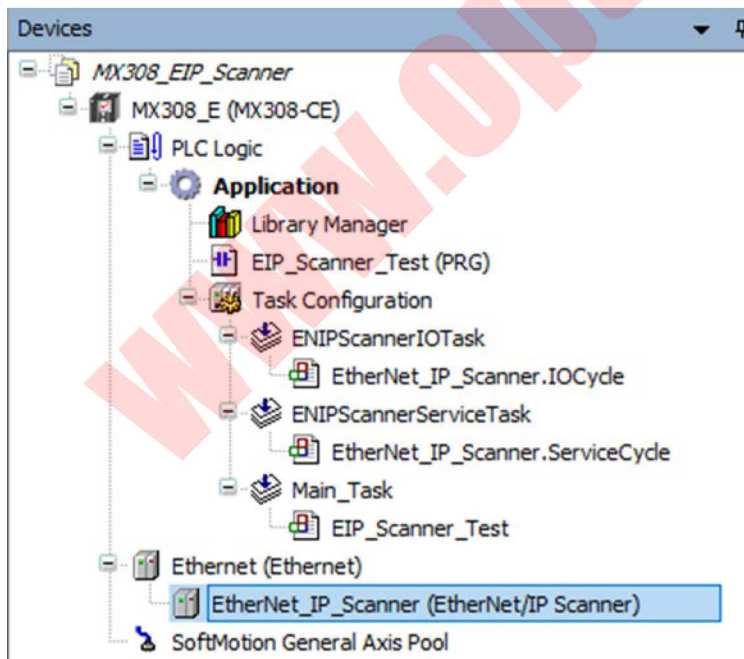
Щёлкните в древе правой кнопкой мышки на пункте **Ethernet** и в отрывшемся меню выберите пункт **Add Device**. Далее в открывшемся окне выберите пункт **Fieldbuses – Ethernet/IP Scanner**:



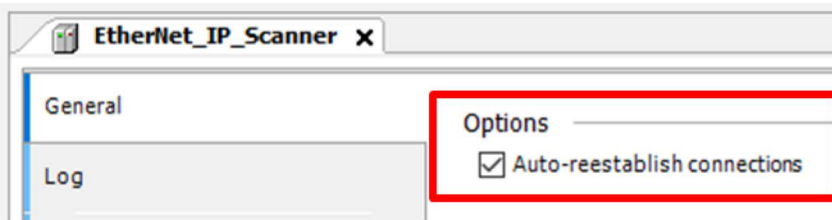
и нажмите **Add Device**:



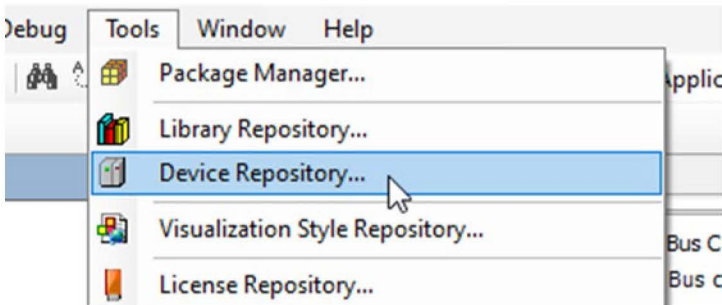
В древе проекта появится пункт **Ethernet/IP Scanner**:



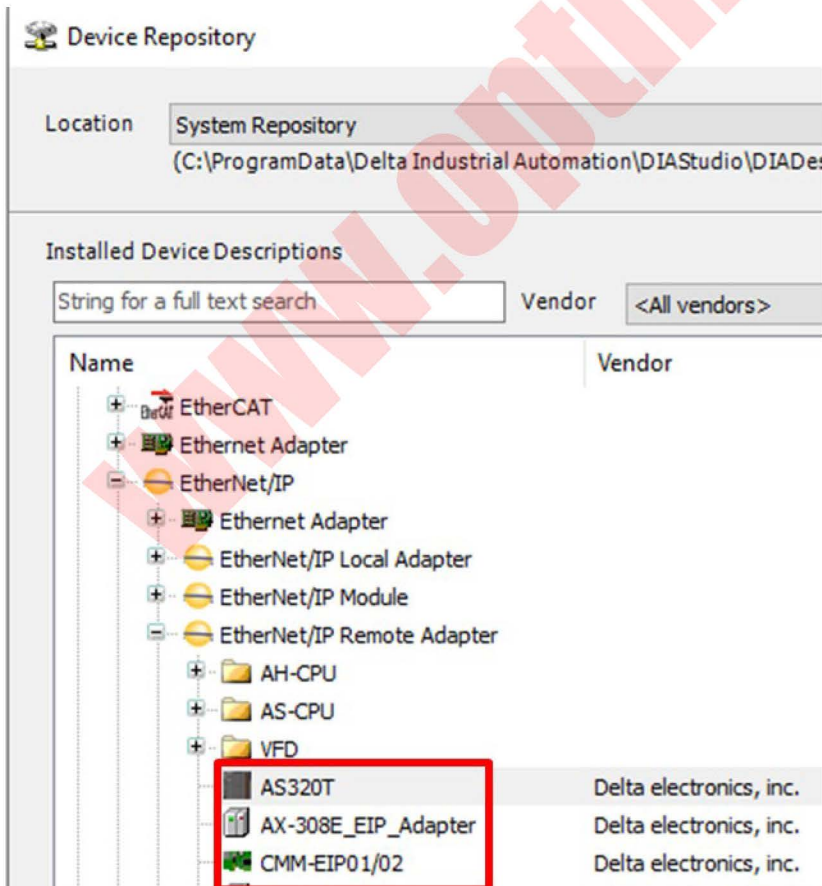
Щёлкните двойным щелчком левой кнопки мышки на пункте **Ethernet/IP Scanner** и в открывшейся вкладке выберите пункт **General**. Установите флажок автоматического восстановления соединения:



Далее можно перейти к добавлению Ведомых устройств (Ethernet/IP Adapter). Перед началом работы необходимо импортировать EDS файл устройства через пункт меню **Tools - Device Repository**:

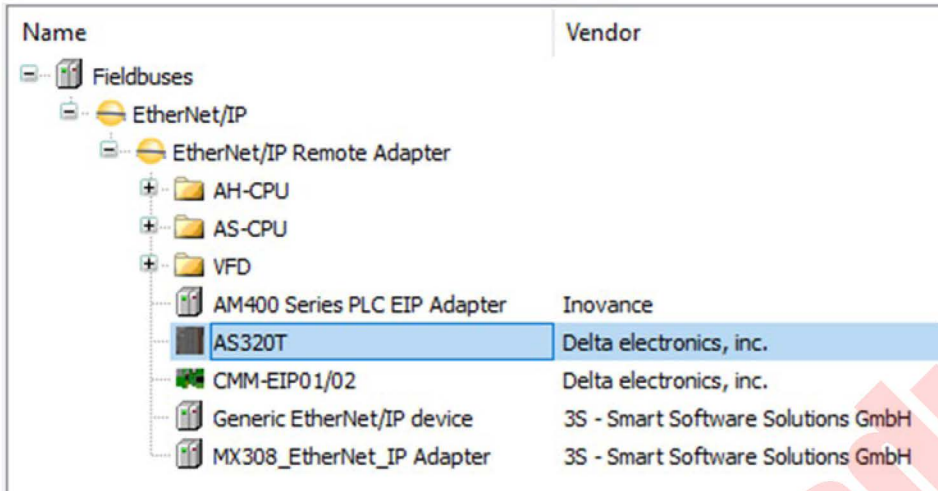


Ниже мы рассмотрим организацию связи с контроллером Delta AS320T и преобразователем частоты Delta MS300 посредством платы CMM-EIP02 по протоколу Ethernet/IP. Предполагается, что на этом этапе будут установлены соответствующие EDS файлы данных устройств:

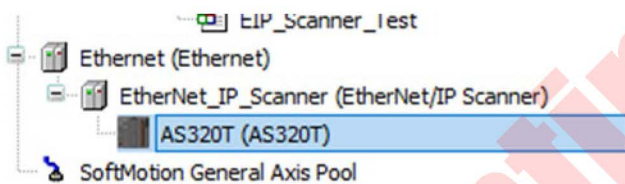


## Пример настройки связи с контроллером Delta AS320T по протоколу Ethernet/IP

Для начала работы необходимо добавить Ведомое устройство в проект. Для этого щёлкните правой кнопкой мышки в древе проекта на пункте **Ethernet/IP Scanner** и в открывшемся меню выберите пункт **Add Device**. В открывшемся окне выберите из **Device Repository** устройство **AS320T**:



В древе проекта появится устройство:



Щёлкните дважды левой кнопкой мышки на пункте AS320T, и в открывшейся вкладке выберите пункт **General** и введите IP адрес ведомого устройства.



Далее в пункте **Connections** можно увидеть разметку данных от Ведомого (AS320T) и к Ведомому от Мастера (MX308), которая будет сделана системой в соответствии с EDS файлом на контроллер AS320T:

AS320T x						
General	Connection Name	RPI (ms)	O-->T Size (Bytes)	T-->O Size (Bytes)	Proxy Config Size (Bytes)	Target Config Size (Bytes)
Connections	Connection1	20	200	200		16

200 байтов от Ведомого (O → T) и 200 к Ведомому (T → O)

Адреса регистров в контроллере AS320T будут размечены также автоматически в соответствии с информацией в EDS файле:

Parameters	Value	Unit	Data Type	Minimum	Maximum	Default
<b>Connection1</b>						
<b>Target Config data</b>						
Conn1_Input(T->O) DeviceType	D		UINT	0	3	0
Conn1_Input(T->O) Reserved	200		UINT	0	500	200
Conn1_Input(T->O) DeviceIndex	1000		UDINT	0	29999	1000
Conn1_Output(O->T) DeviceType	D		UINT	0	3	0
Conn1_Output(O->T) Reserved	200		UINT	0	500	200
Conn1_Output(O->T) DeviceIndex	0		UDINT	0	29999	0

В нашем примере Ведомый контроллер AS320T будет принимать данные в регистры D0 – D99, а отправлять Мастеру из регистров D1000 – D1099. Количество и номера регистров можно поменять.

Далее в пункте **Ethernet/IP I/O Mapping** можно увидеть выделенные системой регистры данных от Ведомого (AS320T) %IW0 - %IW99, и к Ведомому от Мастера (MX308) %QW0 - %QW99, которая будет автоматически сделана системой:

Connections	Variable	Mapping	Channel	Address	Type
<b>Connection1</b>					
	Input_data0			%IW0	UINT
	Input_data1			%IW1	UINT
	Input_data2			%IW2	UINT
	Input_data3			%IW3	UINT
	Input_data4			%IW4	UINT
	Input_data5			%IW5	UINT
	Input_data6			%IW6	UINT
	Input_data7			%IW7	UINT
	Input_data8			%IW8	UINT

Output_data0	%QW0	UINT
Output_data1	%QW1	UINT
Output_data2	%QW2	UINT
Output_data3	%QW3	UINT
Output_data4	%QW4	UINT
Output_data5	%QW5	UINT
Output_data6	%QW6	UINT
Output_data7	%QW7	UINT
Output_data8	%QW8	UINT
Output_data9	%QW9	UINT
Output_data10	%QW10	UINT
Output_data11	%QW11	UINT

Система по умолчанию назначает регистрам Мастер-контроллера MX308 тип данных UINT, т.е. беззнаковый. Но так как в Ведомом контроллере AS320T регистры данных D0 – D29999 являются целочисленными 16 бит со знаком, поэтому в Мастере регистры желательно объявить переменными типа INT:

```

7 |  VAR   EIP_RST_TRG           BOOL
8 |  VAR   _D1000                %IW0    INT

```

Если загрузить проекты и запустить в онлайн режим оба контроллера, то передача данных будет выглядеть следующим образом:

От Ведомого к Мастеру:

В программе Ведомого контроллера AS320T задаются данные в регистрах D1000 и D1099,

Device Name	Status	Data Type	Value (16bits)
D0			0
D99			0
D1000			789
D1099			1234

которые автоматически передаются Мастеру в регистры %IW0 и %IW99:

Watch 1			
Expression	Application	Type	Value
%IW0	MX308_E.Application	WORD	789
%IW99	MX308_E.Application	WORD	1234
%QW0	MX308_E.Application	WORD	0
%QW99	MX308_E.Application	WORD	0
AS320T.eState	MX308_E.Application	ADAPTERSTATE	RUNNING

Состояние связи с Ведомым можно проверить через свойство узла eState:



AS320T.eState	MX308_E.Application	ADAPTERSTATE	RUNNING
---------------	---------------------	--------------	---------

От Мастера к Ведомому:

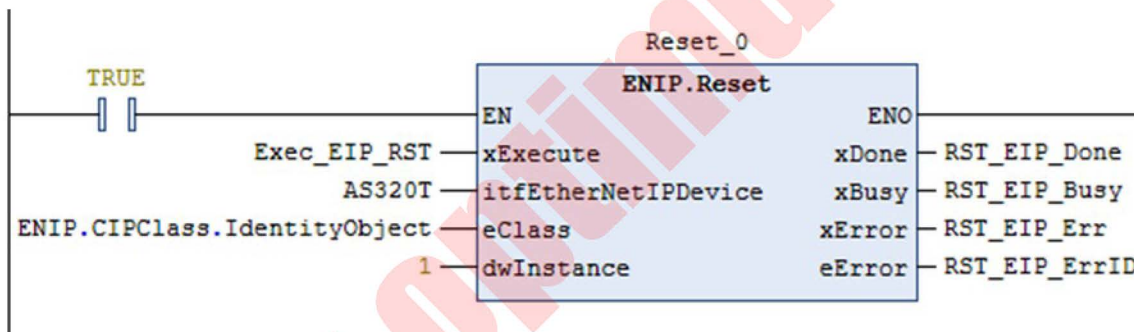
В программе Мастера задаются данные в регистрах %QW0 и %QW99

Watch 1			
Expression	Application	Type	Value
%IW0	MX308_E.Application	WORD	0
%IW99	MX308_E.Application	WORD	0
%QW0	MX308_E.Application	WORD	45
%QW99	MX308_E.Application	WORD	555
AS320T.eState	MX308_E.Application	ADAPTERSTATE	RUNNING

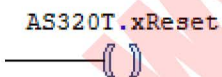
которые автоматически попадают в регистры D0 и D99 Ведомого:

Device Name	Status	Data Type	Value (16bits)
D0			45
D99			555
D1000			0
D1099			0

Для перезагрузки узла можно использовать команду **ENIP.Reset**:



Или свойство узла **xReset**:



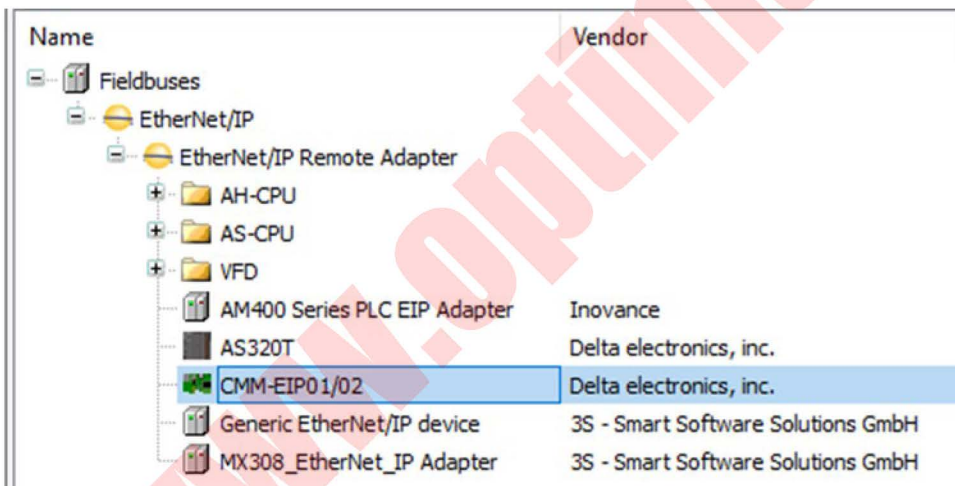
### Пример настройки связи с ПЧ Delta MS300 по протоколу Ethernet/IP

Преобразователи частоты типа Delta MS300 осуществляют связь по протоколу Ethernet/IP посредством коммуникационной платы расширения CMM-EIP02. Перед началом работы необходимо убедиться, что версия встроенного ПО (firmware) платы не ниже 2.05.06, а самого преобразователя частоты не ниже 2.02.

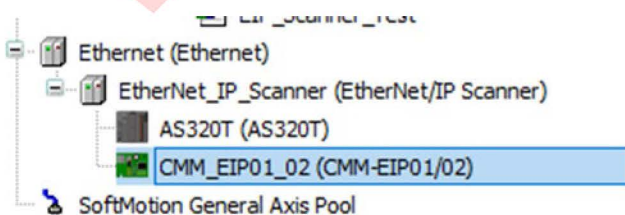
Также, необходимо убедиться, что код продукта в EDS файле на данную плату стоит 17157:

```
[Device]
  VendCode = 799;
  VendName = "Delta electronics, inc.";
  ProdType = 12;
  ProdTypeStr = "Communications Adapter";
  ProdCode = 17157;
  MajRev = 1;
  MinRev = 1;
  ProdName = "CMM-EIP01/02";
  Catalog = "CMM-EIP01/02";
  Icon = "CMMS-EIP01.ico";
  IconContents =
```

Для начала работы необходимо добавить Ведомое устройство в проект. Для этого щёлкните правой кнопкой мышки в древе проекта на пункте **Ethernet/IP Scanner** и в открывшемся меню выберите пункт **Add Device**. В открывшемся окне выберите из **Device Repository** устройство **CMM-EIP01/02**:



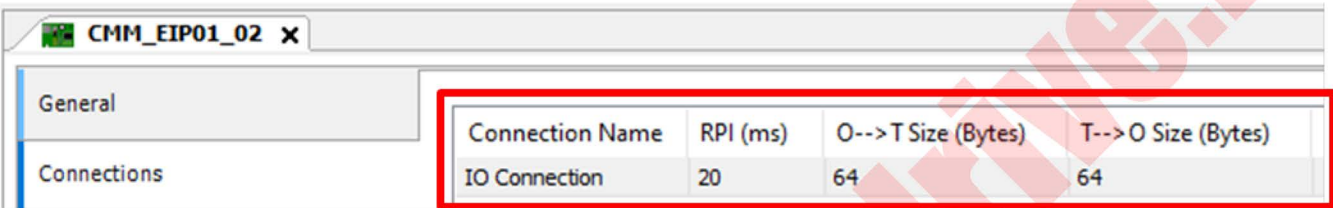
В древе проекта появится устройство:



Щёлкните дважды левой кнопкой мышки на пункте CMM-EIP01/02, и в открывшейся вкладке выберите пункт **General** и введите IP адрес ведомого устройства.



Далее в пункте **Connections** можно увидеть разметку данных от Ведомого (CMM-EIP01/02) и к Ведомому от Мастера (MX308), которая будет сделана системой в соответствии с EDS файлом на плату CMM-EIP01/02:



64 байта от Ведомого (O → T) и 64 к Ведомому (T → O)

Соответствие регистрам привода приводится в таблице ниже:

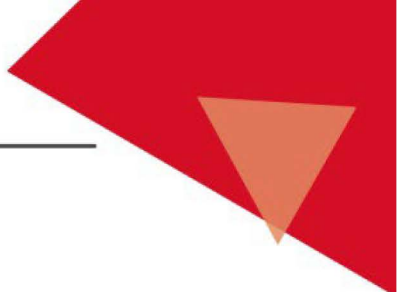
Данные от привода в сторону контроллера:

**Configuration Data**

Raw data values  Show Parameter Groups

Parameters	Value	Unit	Data Type
IN_Addr_1	[2100] Warn / Error		UINT
IN_Addr_2	[2101] Drive status 1		UINT
IN_Addr_3	[2102] Frequency command 1		UINT
IN_Addr_4	[2103] Output frequency		UINT
IN_Addr_5	[2104] Output current		UINT
IN_Addr_6	[2105] DC bus voltage		UINT
IN_Addr_7	[2106] Output voltage		UINT
IN_Addr_8	[2107] Multi-speed step		UINT
IN_Addr_9	[2108] Max output torque		UINT
IN_Addr_10	[2109] Trigger count		UINT
IN_Addr_11	[210A] PF angle		UINT
IN_Addr_12	[210B] Output torque		UINT
IN_Addr_13	[210C] Real speed(rpm)		UINT
IN_Addr_14	[210D] PG feedback count		UINT
IN_Addr_15	[210E] PG2 pulse count		UINT
IN_Addr_16	[210F] Output Power		UINT
IN_Addr_17	[6100] Status word 2		UINT
IN_Addr_18	[6101] Real mode 2		UINT
IN_Addr_19	[6102] Output Frequency 2		UINT

Данные от контроллера к приводе:



Configuration Data

Raw data values  Show Parameter Groups

Parameters	Value	Unit	Data Type
OUT_Addr_1	[2000] Operation Command 1		UINT
OUT_Addr_2	[2001] Frequency command 1		UINT
OUT_Addr_3	[2002] External Command 1		UINT
OUT_Addr_4	Reserved		UINT
OUT_Addr_5	[6000] Operation Command2		UINT
OUT_Addr_6	[6001] Control mode 2		UINT
OUT_Addr_7	[6002] Frequency command 2		UINT
OUT_Addr_8	[6003] Troque limit 2		UINT
OUT_Addr_9	[6004] Position Command 2, L W		UINT
OUT_Addr_10	[6005] Position Command 2, H W		UINT
OUT_Addr_11	[6006] Torque command 2		UINT
OUT_Addr_12	[6007] Frequency limit 2		UINT

Далее в пункте **Ethernet/IP I/O Mapping** можно увидеть выделенные системой регистры данных от Ведомого (MS300) %IW100- %IW131, и к Ведомому от Мастера (MX308) %QW100 - %QW131, которая будет автоматически сделана системой:

IO Connection

+	Error Code	%IW 100	UINT	
+	Reserved2101	%IW 101	UINT	
+	FREQ_SET	Frequency Command	%IW 102	UINT Hz
+	OUT_FREQ	Output Frequency	%IW 103	UINT Hz
+	DC_BUS	Output Current	%IW 104	UINT A
+		DC Bus Voltage	%IW 105	UINT V
+		Output Voltage	%IW 106	UINT V
+		Speed	%IW 107	UINT
+		Control	%QW 100	UINT
+		Frequency Command	%QW 101	UINT Hz
+		Others	%QW 102	UINT
+		OUT_Value	%QW 103	UINT
+	CTRL_WORD	OUT_Value	%QW 104	UINT
+		OUT_Value	%QW 105	UINT
+	FREQ_CMD	OUT_Value	%QW 106	UINT
+		OUT_Value	%QW 107	UINT
+		OUT_Value	%QW 108	UINT

Далее регистрам можно присвоить переменные и использовать в программе контроллера для управления приводом:

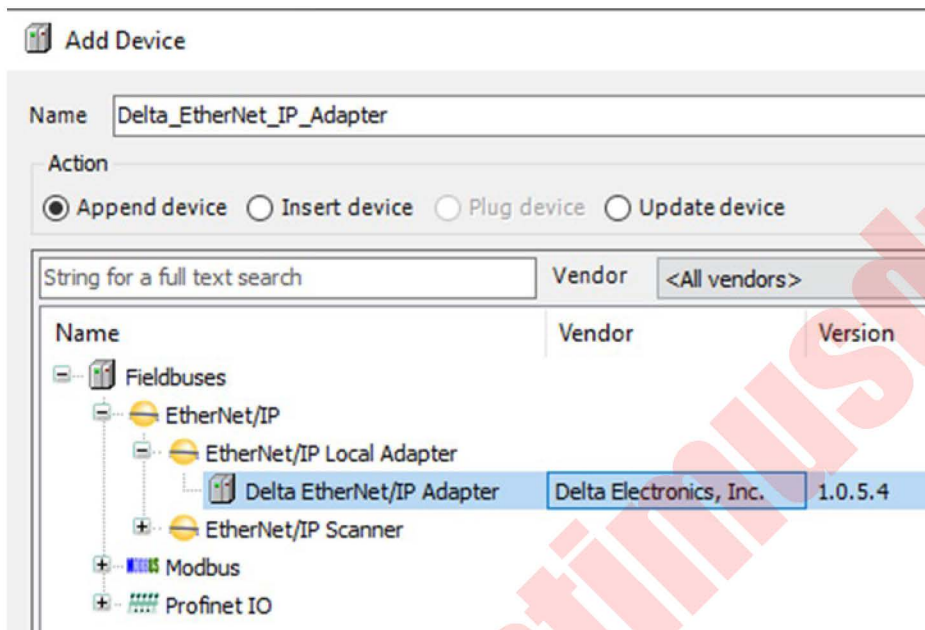
CMM_EIP01_02.eState	MX308_E.Application	ADAPTERSTATE	RUNNING
CTRL_WORD	MX308_E.Application	UINT	129
FREQ_CMD	MX308_E.Application	UINT	2500
FREQ_SET	MX308_E.Application	UINT	2500
OUT_FREQ	MX308_E.Application	UINT	1154
DC_BUS	MX308_E.Application	UINT	3060
STATUS_WORD	MX308_E.Application	UINT	13571

## Пример настройки связи с другим контроллером CODESYS по протоколу Ethernet/IP

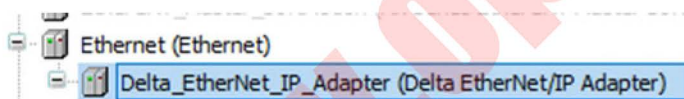
В среде программирования CODESYS есть специальная процедура для настройки связи между двумя контроллерами CODESYS по протоколу Ethernet/IP. В нашем примере в качестве Скенера (Мастера) будет выступать контроллер MX308, а в качестве Адаптера (Ведомого) контроллер Delta AX-308E.

Создайте проект для контроллера Delta AX-308E. При необходимости воспользуйтесь Руководством на данный контроллер.

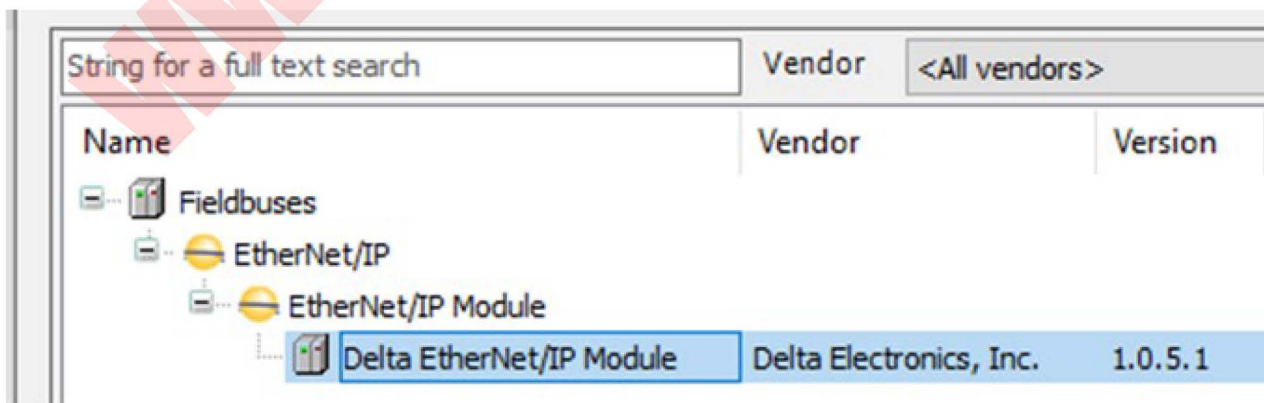
Добавьте в проект адаптер Ethernet. Выделите мышкой этот пункт в древе проекта и нажмите правую кнопку мышки. В появившемся меню выберите пункт **Add Device**. В открывшемся окне выберите **Delta Ethernet/IP Adapter**:



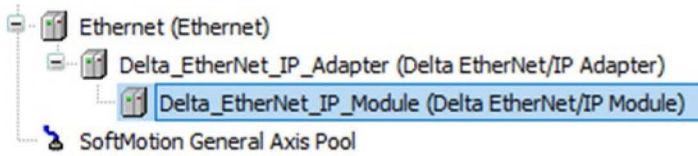
В древе проекта появится пункт:



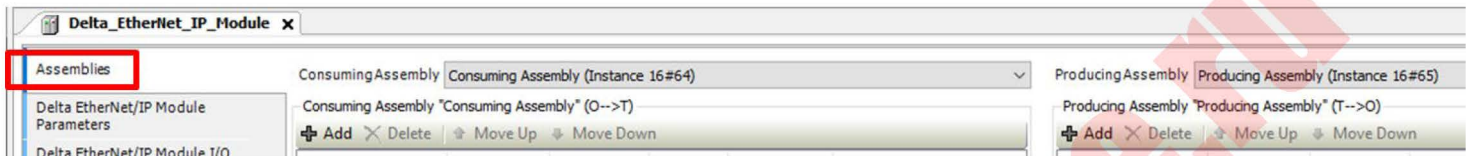
Встаньте мышкой на этот пункт и щёлкните правой кнопкой. В появившемся меню выберите пункт **Add Device**. В открывшемся окне выберите **Delta Ethernet/IP Module**:



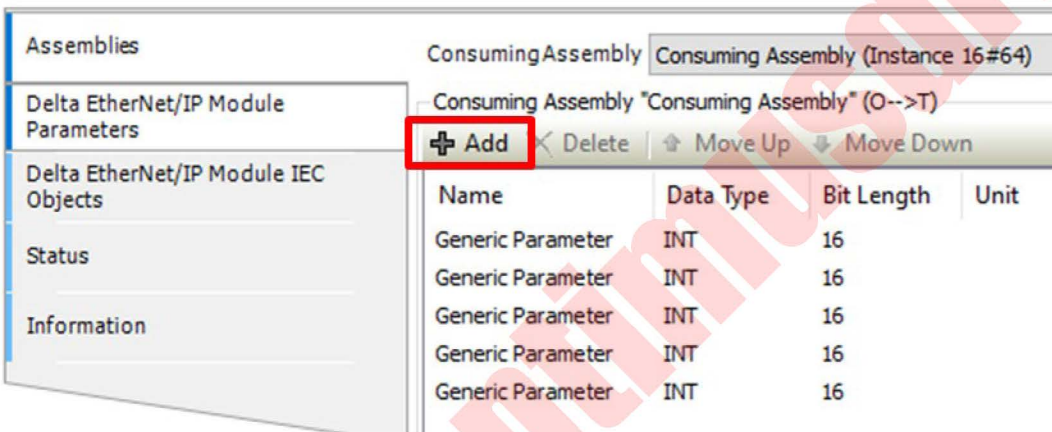
В древе проекта появится пункт:



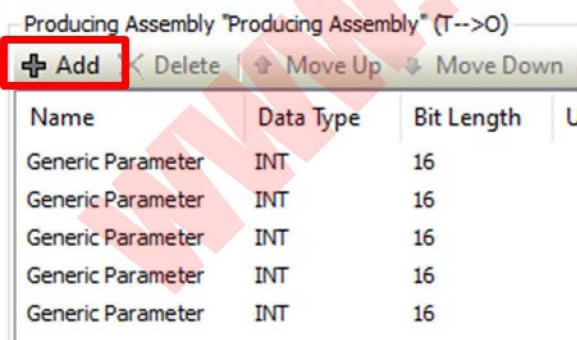
Щёлкните на этом пункте дважды левой кнопкой мышки и в отрывшейся вкладке выберите раздел **Assemblies**:



В данном разделе создаются пакеты для отправки и приёма от Мастера. Для добавления регистров в пакет на приём от Мастера нажимайте кнопку **Add** в поле **Consuming Assembly**:



Для добавления регистров в пакет на отправку Мастеру нажимайте кнопку **Add** в поле **Producing Assembly**:



Общий список регистров можно увидеть в разделе **Delta Ethernet/IP Module I/O Mapping**. Здесь же можно присвоить регистрам теги. Регистры %IW2-%IW6 будут содержать данные от Мастера (Consumed Tags), а из регистров %QW1-%QW5 данные будут передаваться Мастеру (Produced Tags).

Assemblies	Find	Filter	Show all			
	Variable	Mapping	Channel	Address	Type	Unit
Delta EtherNet/IP Module Parameters	+ [icon]		Generic Parameter	%IW2	INT	
Delta EtherNet/IP Module I/O Mapping	+ [icon]		Generic Parameter	%IW3	INT	
Delta EtherNet/IP Module IEC Objects	+ [icon]		Generic Parameter	%IW4	INT	
Status	+ [icon]		Generic Parameter	%IW5	INT	
Information	+ [icon]		Generic Parameter	%IW6	INT	
	+ [icon]		Generic Parameter	%QW1	INT	
	+ [icon]		Generic Parameter	%QW2	INT	
	+ [icon]		Generic Parameter	%QW3	INT	
	+ [icon]		Generic Parameter	%QW4	INT	
	+ [icon]		Generic Parameter	%QW5	INT	

Для экспорта созданной таблицы данных для обмена с Мастером необходимо двойным щелчком открыть вкладку **Delta Ethernet/IP Adapter** и выбрать раздел **Delta Ethernet/IP Adapter I/O Mapping**:

Нажмите кнопку **Export EDS File** и получите XML файл с тегами. Лучше изменить **ProdName** со стандартного на своё название. Это поможет избежать путаницы со стандартным адаптером при выборе устройства из репозитория.

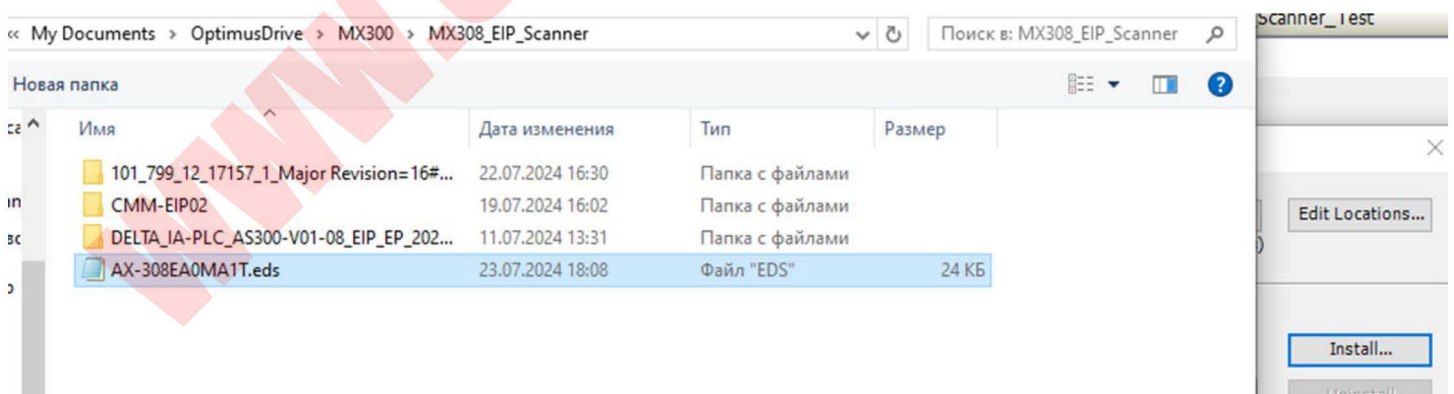
Для изменения поля откройте файл в Блокноте (Notepad) и измените название поле **ProdName**

```
[Device]
VendCode = 799;
VendName = "Delta electronics, inc.";
ProdType = 12;
ProdTypeStr = "Communications Adapter";
ProdCode = 16386;
MajRev = 1;
MinRev = 1;
ProdName = "AX-308EA0MA1T";
```

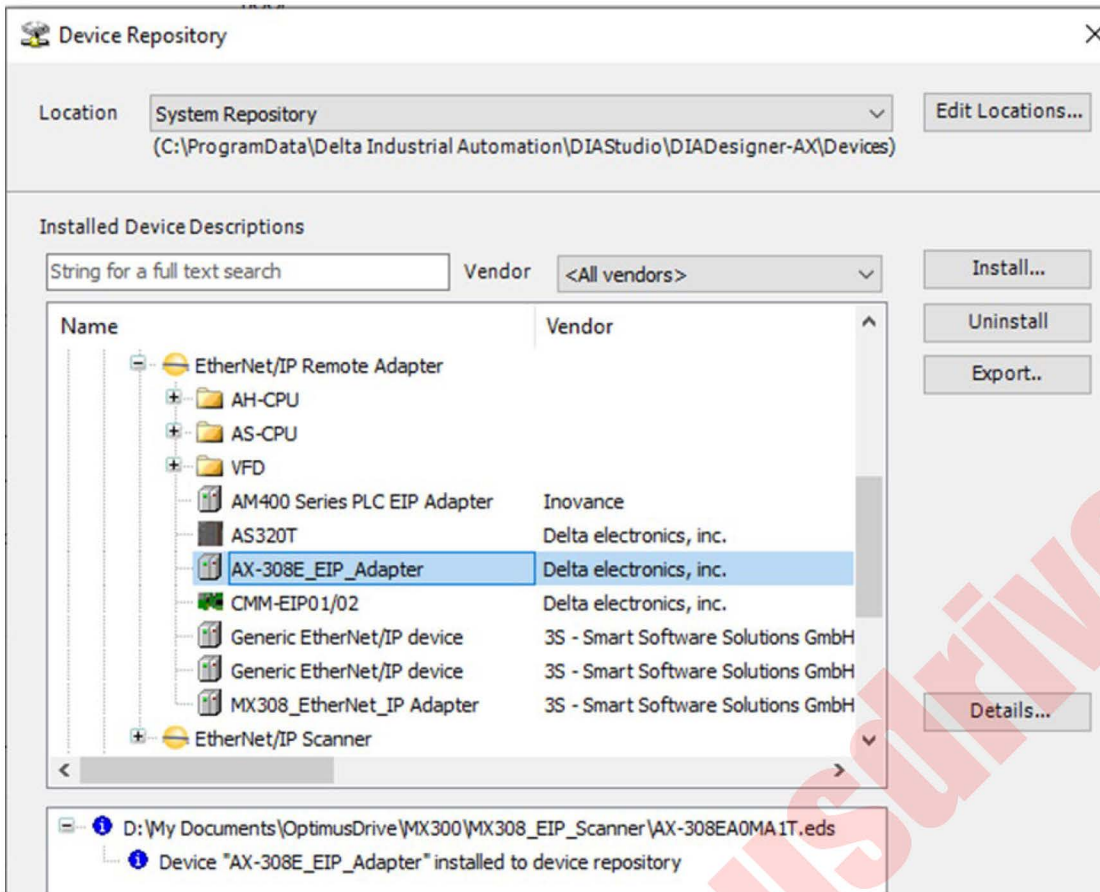
например на такое:

```
[Device]
VendCode = 799;
VendName = "Delta electronics, inc.";
ProdType = 12;
ProdTypeStr = "Communications Adapter";
ProdCode = 16386;
MajRev = 1;
MinRev = 1;
ProdName = "AX-308E_EIP_Adapter";
```

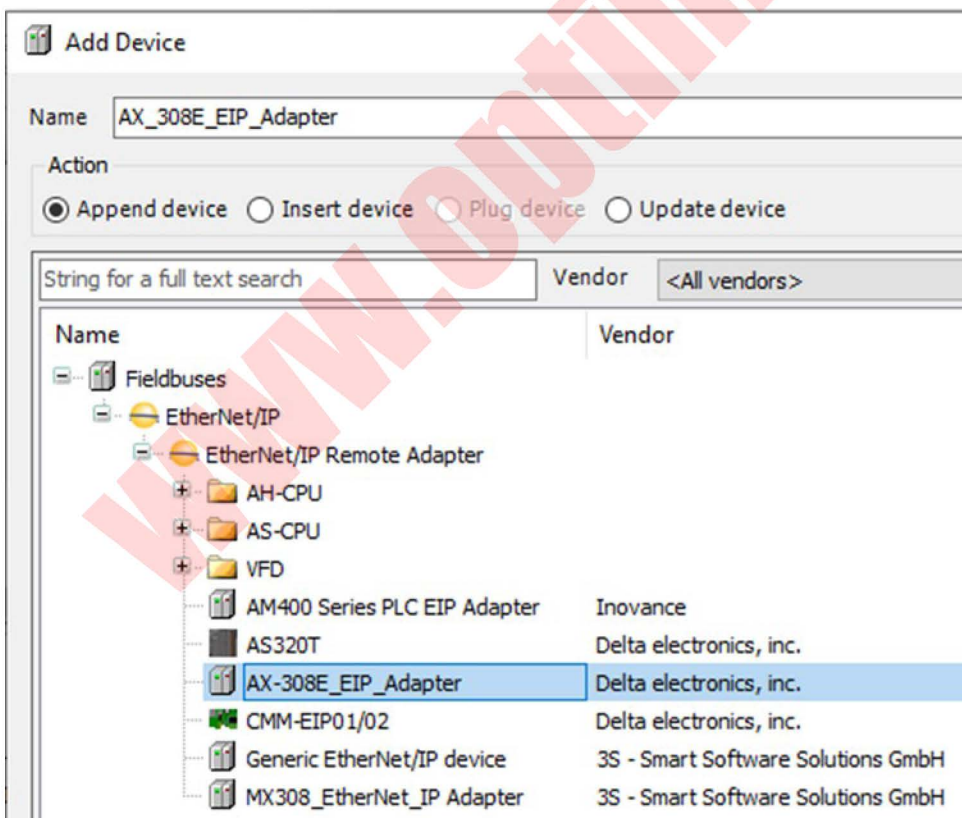
Далее файл необходимо импортировать в репозиторий устройств Мастера (MX300). Т.е. там появится устройство типа Adapter Ethernet/IP. Данная процедура, как и создание проекта для контроллеров MX300, описаны в соответствующих Главах настоящего Руководства.



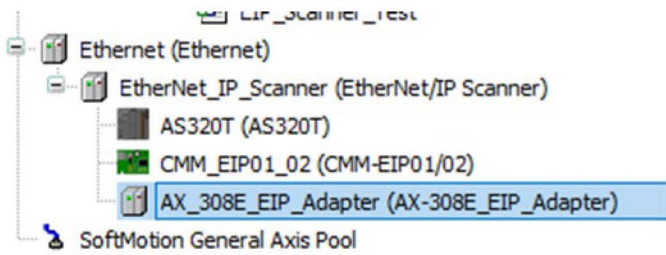




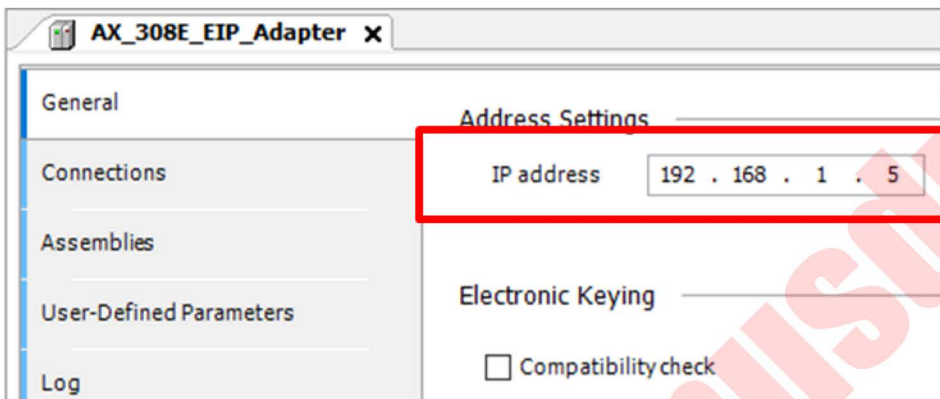
Добавьте адаптер в проект Мастера:



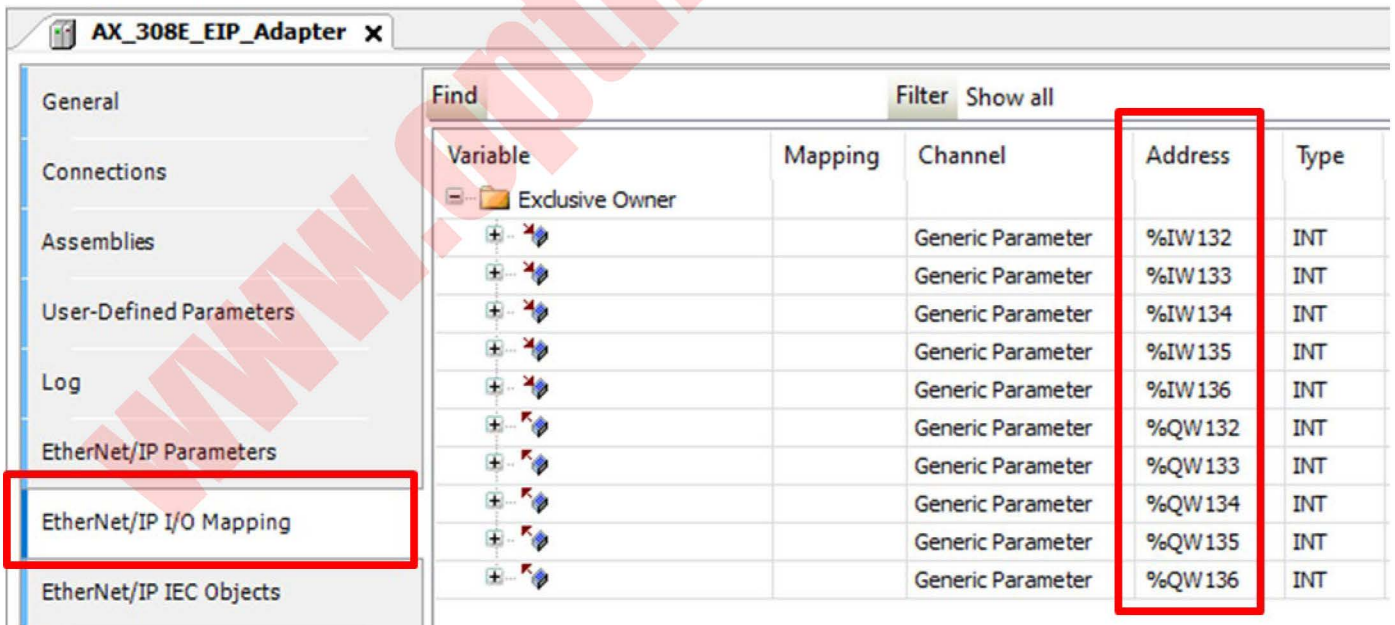
В древе проекта появится устройство:



Выберите это устройство и щёлкните дважды левой кнопкой мышки. Откроется вкладка с настройками. Выберите пункт General и выставьте IP адрес EIP Adapter (AX308-E):



Система автоматически раздаст адреса в соответствии с импортированным XML файлом, которые можно посмотреть в пункте **EtherNet/IP I/O Mapping**:



Регистры %IW132-%IW136 будут содержать данные от Ведомого (Produced Tags), а из регистров %QW132-%QW136 данные будут передаваться Ведомому (Consumed Tags). (Produced Tags и Consumed Tags с точки зрения Ведомого).

Если в Мастере (контроллер MX308) в регистре %QW132 задать число 45:

%IW132	MX308_E.Application	WORD	0
%QW132	MX308_E.Application	WORD	45
AX_308E_EIP_Adapter.eState	MX308_E.Application	ADAPTERSTATE	RUNNING

То в регистре %IW2 Ведомого (контроллер Delta AX-308E) появится число 45:

Watch 1			
Expression	Application	Type	Value
%IW2	Device.Application	WORD	45
%QW1	Device.Application	WORD	0
Delta_EtherNet_IP_Module.eState	Device.Application	MODULESTATE	RUNNING

Состояние связи можно контролировать при помощи элемента структуры Adapter\_Name.eState

AX\_308E\_EIP\_Adapter.eState

Delta\_EtherNet\_IP\_Module.eState

Если в регистре Ведомого (Delta AX-308E) %QW1 задать число 555:

Watch 1			
Expression	Application	Type	Value
%IW2	Device.Application	WORD	0
%QW1	Device.Application	WORD	555
Delta_EtherNet_IP_Module.eState	Device.Application	MODULESTATE	RUNNING

в регистре Мастера (MX308) %IW132 появится число 555:

%IW132	MX308_E.Application	WORD	555
%QW132	MX308_E.Application	WORD	0
AX_308E_EIP_Adapter.eState	MX308_E.Application	ADAPTERSTATE	RUNNING

## Связь по протоколу Ethernet/IP Adapter (Slave)

Протокол **Ethernet/IP** (IP = Industrial Protocol) является одним из наиболее распространённых промышленных протоколов в мире и поддерживается сотнями производителей промышленного оборудования. **Ethernet/IP** (сокращённо **EIP**) использует модель **CIP** (Common Industrial Protocol) поверх стандартного интерфейса **Ethernet** и работает по схеме **Producer/Consumer**. На транспортном уровне используется протокол **UDP**.

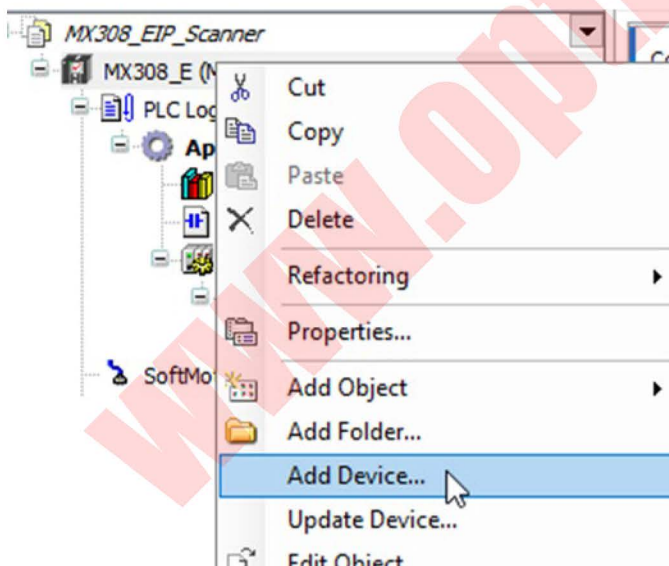
Контроллеры серии MX300 могут работать как в режиме **Ethernet/IP Scanner** (Master), так и в режиме **Ethernet/IP Adapter** (Slave). В данной главе рассматривается организация связи контроллера MX300 с Мастером в режиме Ethernet/IP Adapter (Slave).

Для использования Ethernet/IP к проекту должны быть подключены следующие библиотеки: IoDrvEthernet, IoDrvEtherNetIP, EtherNetIP Services и DED.

CAA Device Diagnosis = CAA Device Diagnosis, 3.5.15.0 (CAA Technical Workgroup)	DED	3.5.15.0
EtherNetIP Services = EtherNetIP Services, 4.5.0.0 (3S - Smart Software Solutions GmbH)	ENIP	4.5.0.0
IoDrvEthernet = IoDrvEthernet, 4.2.0.0 (3S - Smart Software Solutions GmbH)	IoDrvEthernet	4.2.0.0
IoDrvEtherNetIP = IoDrvEtherNetIP, 4.5.1.0 (3S - Smart Software Solutions GmbH)	IoDrvEtherNetIP	4.5.1.0

Настройка связи по протоколу Ethernet/IP начинается с добавления к проекту адаптера Ethernet и привязка его к определённому порту.

Щёлкните в древе правой кнопкой мышки на пункте **Device (MX308\_E)** и в отрывшемся меню выберите пункт **Add Device**



В открывшемся окне выберите **Fieldbuses – Ethernet**:

Add Device

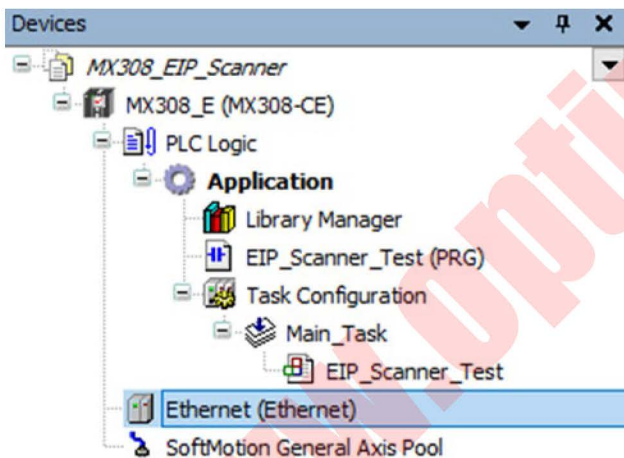
Name

Action  
 Append device  
  Insert device  
  Plug device  
  Update device

String for a full text search  Vendor

Name	Vendor	Version	Description
Miscellaneous			
Delta Localbus Master			
Fieldbuses			
CANbus			
EtherCAT			
Ethernet Adapter			
<b>Ethernet</b>	CODESYS	4.2.0.0	Ethernet Link.
EtherNet/IP			
Home&Building Automation			
Modbus			
Profinet IO			

В древе проекта появится пункт **Ethernet**:



Щёлкните на нём дважды левой кнопкой мышки и в открывшейся вкладке выберите пункт **General**:

**Ethernet** x

General

Ethernet Device Parameters

Ethernet Device I/O Mapping

Ethernet Device IEC Objects

Network interface  Browse...

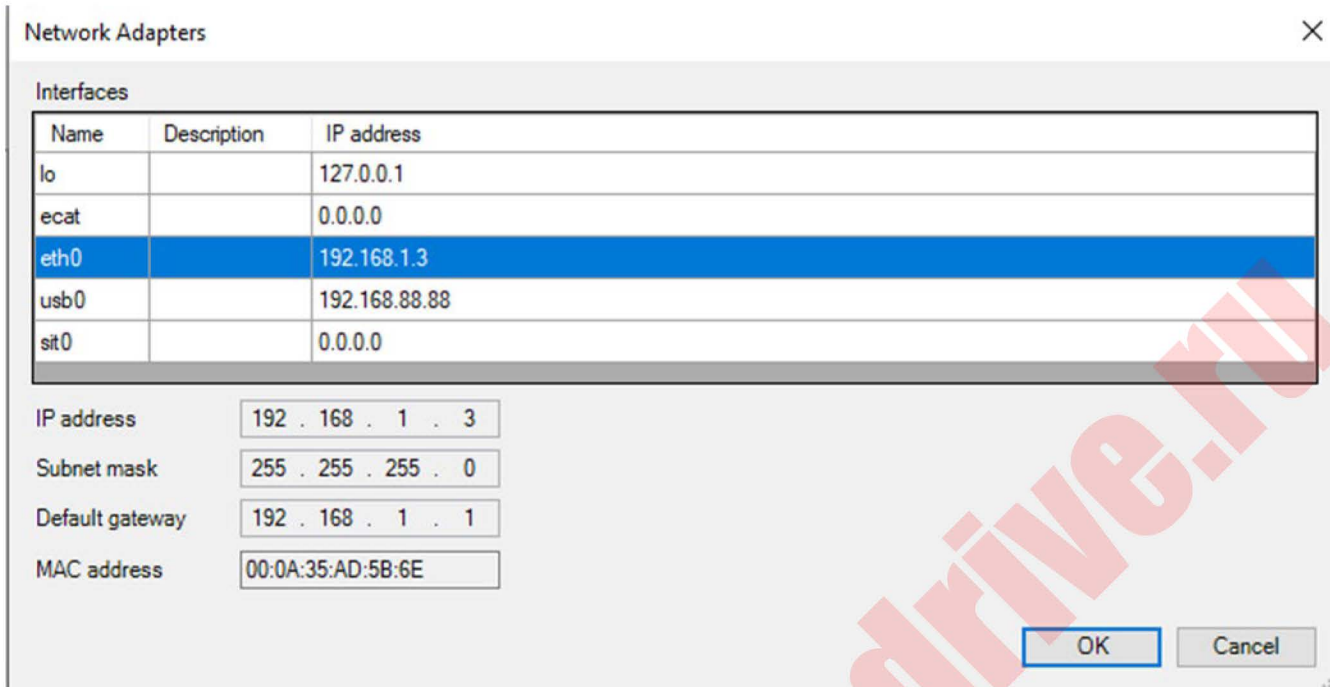
IP address

Subnet mask

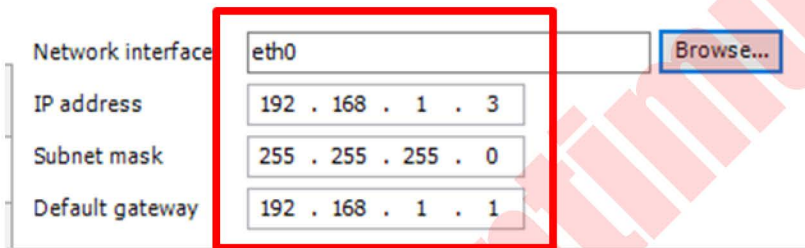
Default gateway

Adjust operating system settings

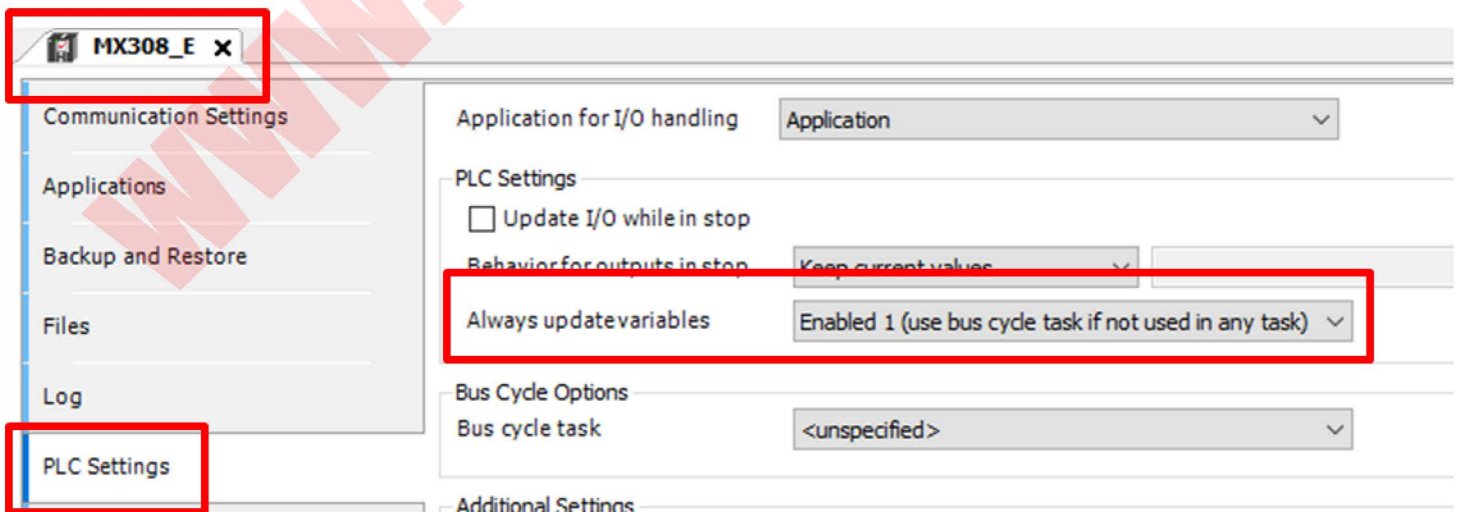
Нажмите кнопку **Browse** и в открывшемся окне выберите IP адрес контроллера:



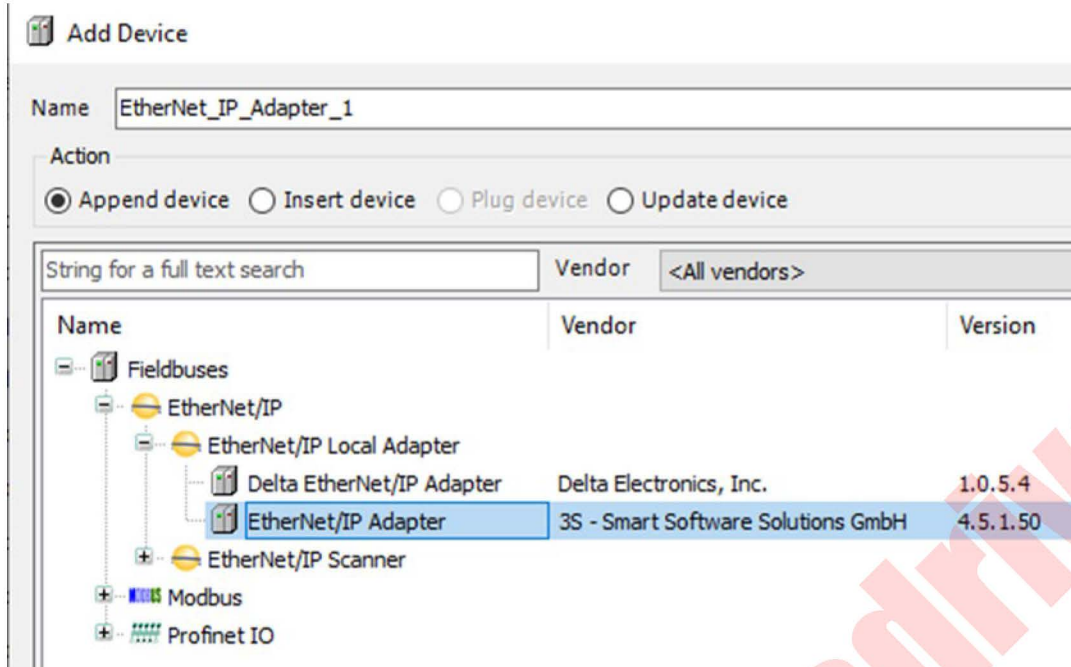
После чего Адаптер будет привязан к Ethernet порту с конкретным IP адресом:



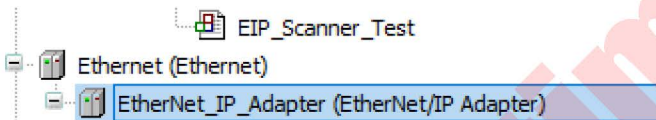
Далее двойным щелчком левой кнопки мыши откройте вкладку **Device** и в пункте **PLC Settings** разрешите обновление переменных. Данный шаг позволяет видеть изменение переменных и регистров без написания в теле программы, т.е. в таблице Watch и в таблицах мэпинга.



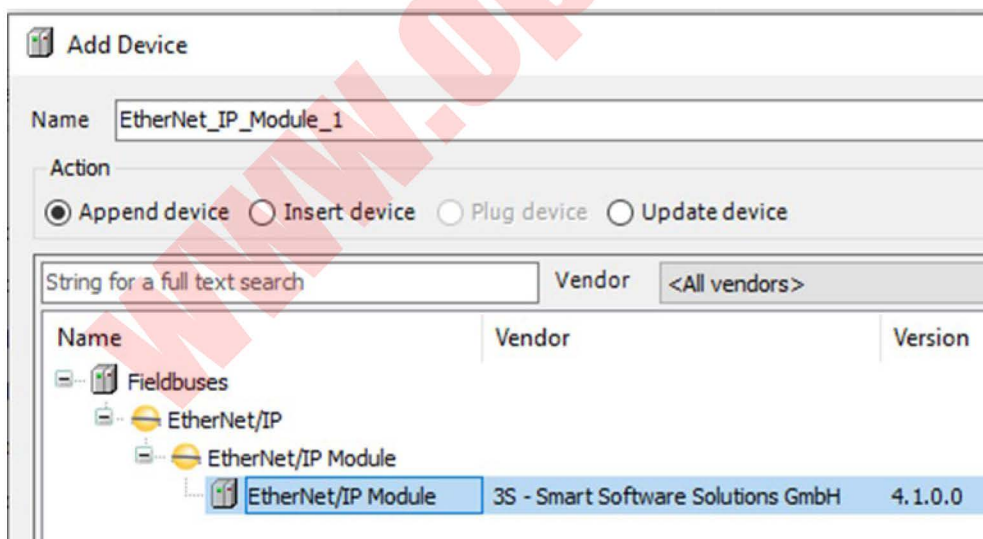
Щёлкните в древе правой кнопкой мышки на пункте **Ethernet** и в отрывшемся меню выберите пункт **Add Device**. Далее в открывшемся окне выберите пункт **Fieldbuses – Ethernet/IP Adapter**:



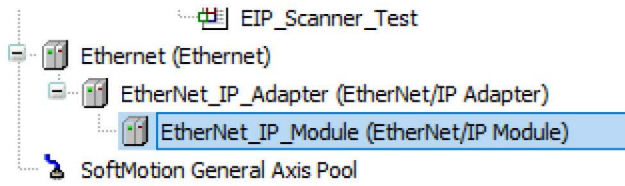
В древе проекта появится устройство:



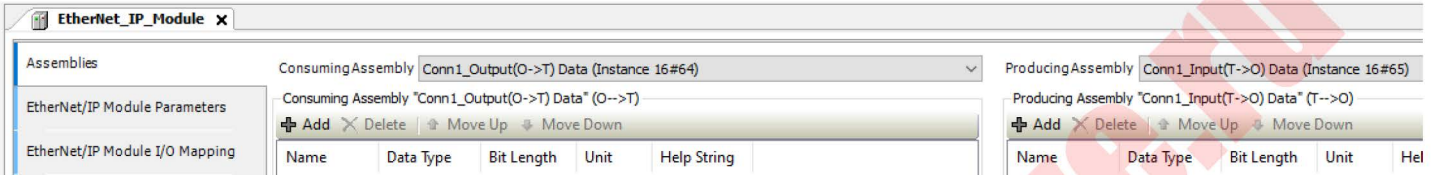
Щёлкните правой кнопкой мышки на данном пункте и добавьте устройство типа **Ethernet/IP Module**:



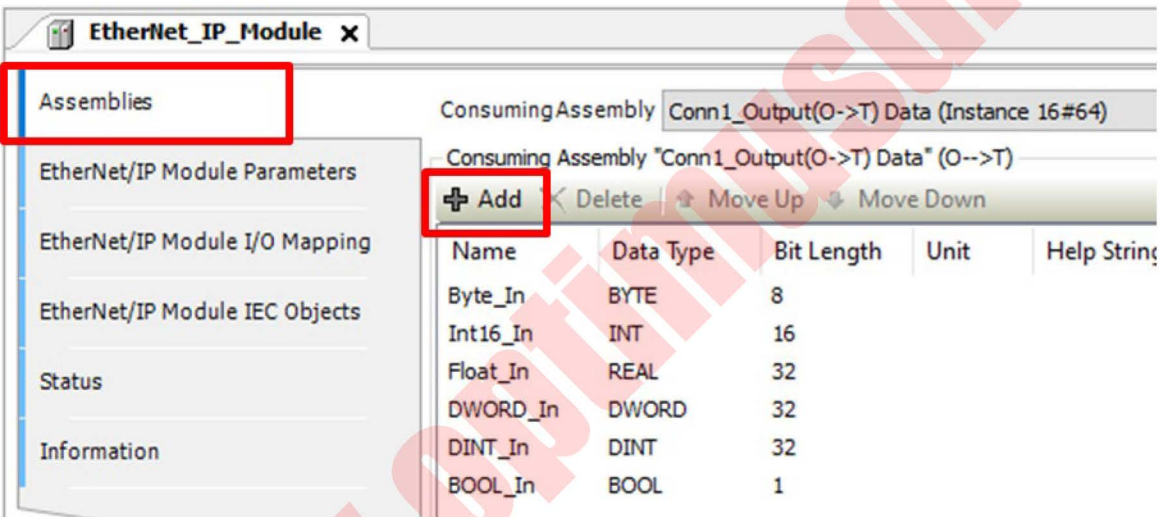
В древе проекта появится устройство:



Щёлкните дважды левой кнопкой мышки на данном пункте и в открывшейся вкладке выберите пункт **Assemblies**:

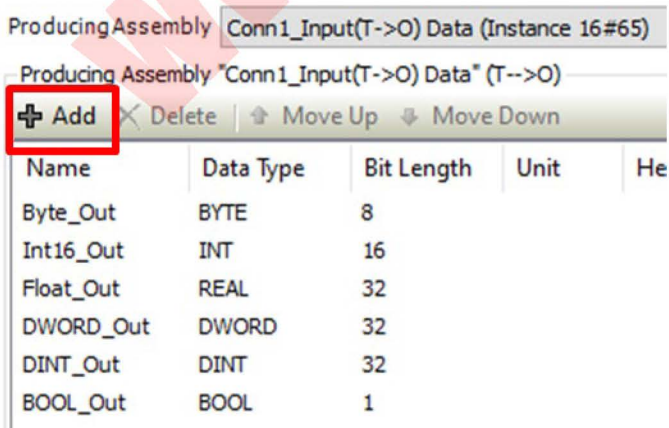


В данном разделе создаются пакеты для отправки и приёма от Мастера. Для добавления регистров в пакет на приём от Мастера нажимайте кнопку **Add** в поле **Consuming Assembly**:



Теги можно создавать сразу с названием и любого из поддерживаемых типов данных.

Для добавления регистров в пакет на отправку Мастеру нажимайте кнопку **Add** в поле **Producing Assembly**:





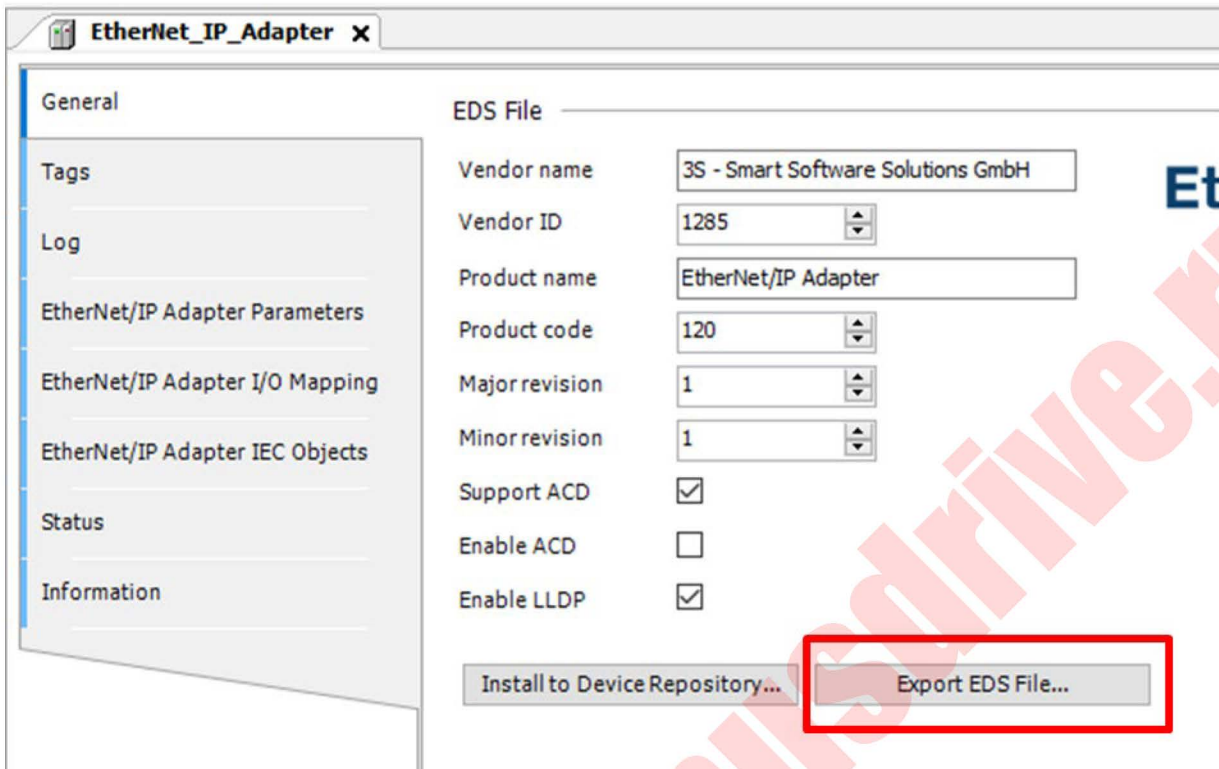
Общий список регистров можно увидеть в разделе **EtherNet/IP Module I/O Mapping**. Здесь же можно присвоить регистрам теги. Регистры типа %I\*\* будут содержать данные от Мастера (Consuming Tags), а из регистров типа %Q\*\* данные будут передаваться Мастеру (Producing Tags).

Variable	Mapping	Channel	Address	Type	Unit
+		Byte_In	%IB0	BYTE	
+		Int16_In	%IW1	INT	
+		Float_In	%ID1	REAL	
+		DWORD_In	%ID2	DWORD	
+		DINT_In	%ID3	DINT	
+		BOOL_In	%IX16.0	BOOL	
+		Byte_Out	%QB0	BYTE	
+		Int16_Out	%QW1	INT	
+		Float_Out	%QD1	REAL	
+		DWORD_Out	%QD2	DWORD	
+		DINT_Out	%QD3	DINT	
+		BOOL_Out	%QX16.0	BOOL	

Для удобства лучше сразу присвоить тегам имена, которые можно будет использовать непосредственно в программе как Ведомого устройства (MX300):

Variable	Mapping	Channel	Address	Type	Unit
+		Byte_In	%IB0	BYTE	
+		Int16_In	%IW1	INT	
+		Float_In	%ID1	REAL	
+		DWORD_In	%ID2	DWORD	
+		DINT_In	%ID3	DINT	
+		BOOL_In	%IX16.0	BOOL	
+		Byte_Out	%QB0	BYTE	
+		Int16_Out	%QW1	INT	
+		Float_Out	%QD1	REAL	
+		DWORD_Out	%QD2	DWORD	
+		DINT_Out	%QD3	DINT	
+		BOOL_Out	%QX16.0	BOOL	

Для экспорта таблицы тегов в виде стандартизированного XML файла необходимо дважды щёлкнуть левой кнопкой мышки на устройстве **EtherNet/IP\_Adapter** и в открывшейся вкладке выбрать пункт **General**:



Нажмите кнопку **Export EDS File** и получите XML файл с тегами.

В файле лучше изменить поле **ProdName** со стандартного на своё название. Это поможет избежать путаницы со стандартным адаптером при выборе устройства из репозитория.

Для изменения поля откройте файл в Блокноте (Notepad) и измените название поле **ProdName**

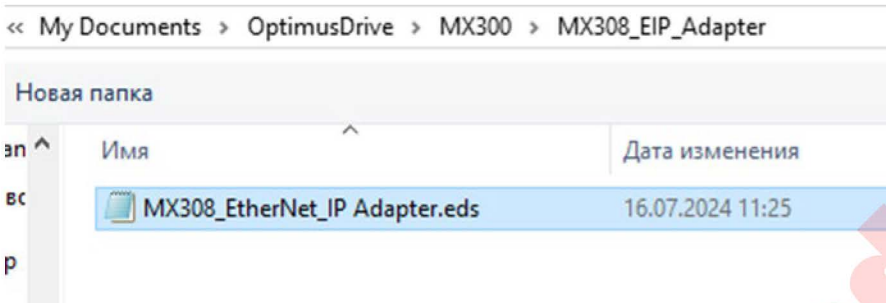
```
[Device]
VendCode = 1285;
VendName = "3S - Smart Software Solutions GmbH";
ProdType = 12;
ProdTypeStr = "Communications Adapter";
ProdCode = 120;
MajRev = 1;
MinRev = 1;
ProdName = "EtherNet/IP Adapter";
```

например на такое:

[Device]

```
VendCode = 1285;
VendName = "3S - Smart Software Solutions GmbH";
ProdType = 12;
ProdTypeStr = "Communications Adapter";
ProdCode = 120;
MajRev = 1;
MinRev = 1;
ProdName = "MX308_EtherNet_IP Adapter";
```

И можно изменить имя файла:

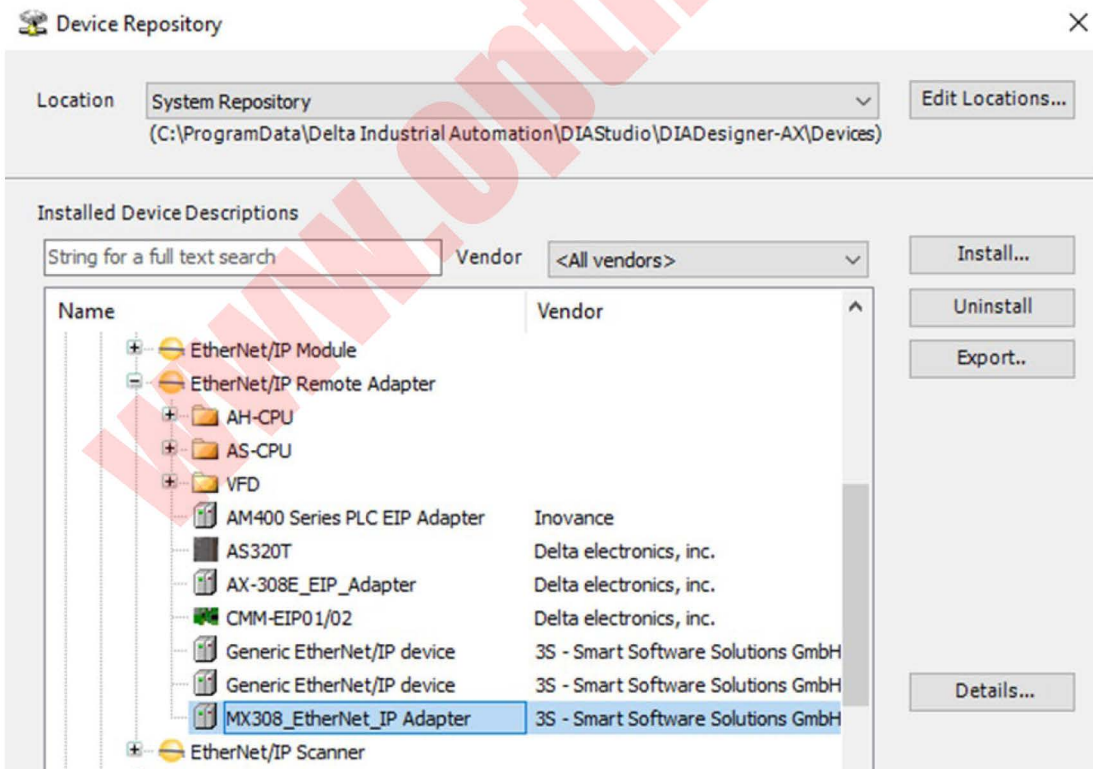


Данный файл можно импортировать в любой **Ethernet/IP Scanner (Master)**.

В качестве **Ethernet/IP Scanner (Master)** рассмотрим контроллер Delta AX-308E.

Создайте проект. При необходимости используйте документацию на данный контроллер.

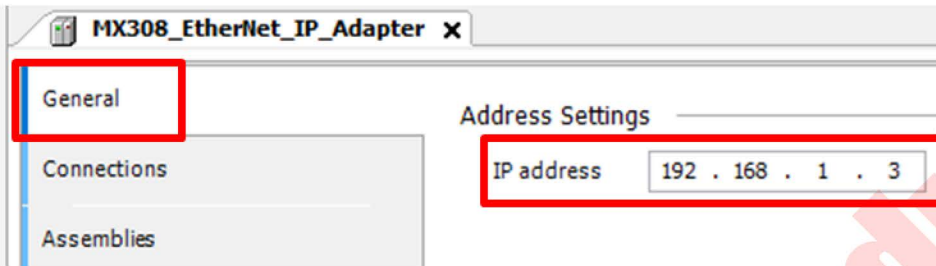
Импортируйте в репозиторий устройств полученный на предыдущем шаге XML файл с тегами от контроллера MX308 (Adapter).



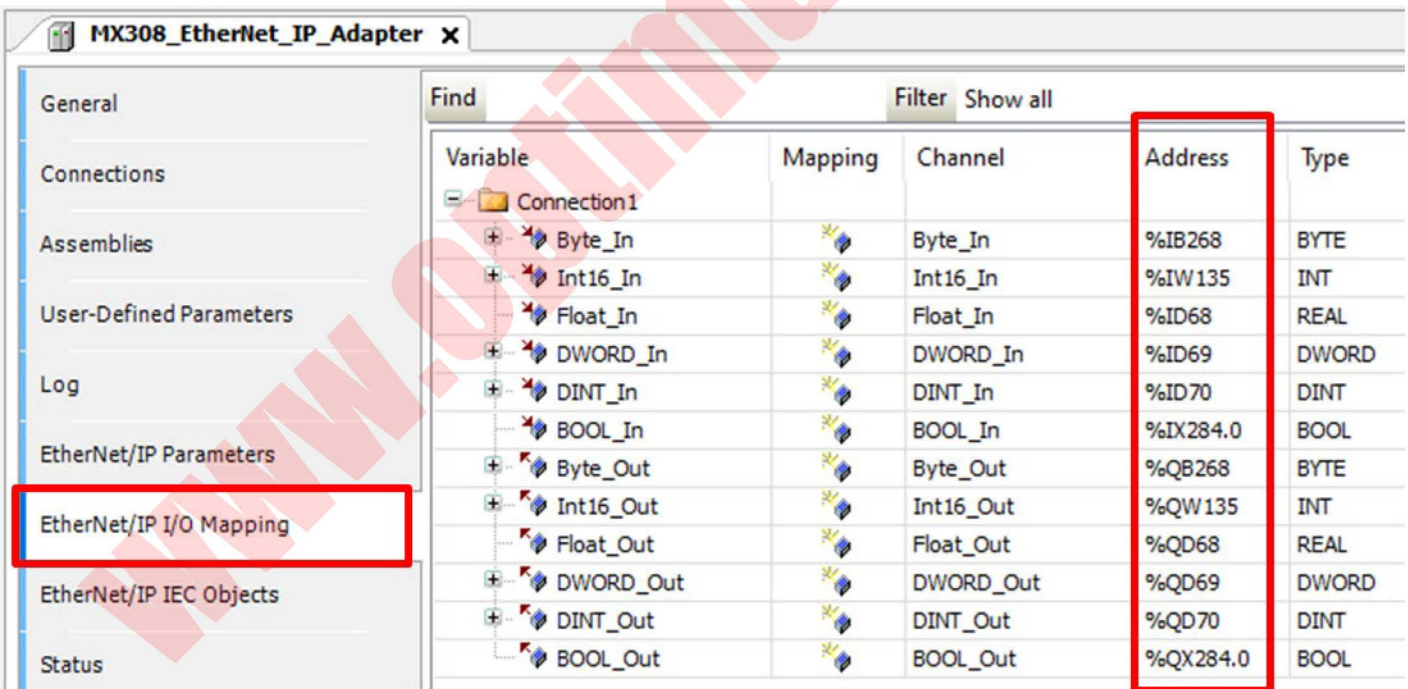
Добавьте устройство в проект:



Щёлкните на нём двойным щелчком левой кнопкой мышки и в отрывшейся вкладке выберите пункт **General**, где необходимо установить IP адрес Адаптера:



Система автоматически раздаст адреса тегам, которые содержались в импортированном XML файле Адаптера (контроллера MX300). Адреса можно посмотреть в пункте **EtherNet/IP I/O Mapping**:



Также, необходимо объявить переменные для обращения к импортированным тегам в программе Мастера. Это является обязательным, так как при обращении к физическим регистрам они трактуются как целочисленные. А при объявлении переменной тип данных объявляется явным образом и система будет правильно трактовать данные:














Variable	Mapping	Channel	Address	Type
Connection1				
Byte_In		Byte_In	%IB268	BYTE
Int16_In		Int16_In	%IW135	INT
Float_In		Float_In	%ID68	REAL
DWORD_In		DWORD_In	%ID69	DWORD
DINT_In		DINT_In	%ID70	DINT
BOOL_In		BOOL_In	%IX284.0	BOOL
Byte_Out		Byte_Out	%QB268	BYTE
Int16_Out		Int16_Out	%QW135	INT
Float_Out		Float_Out	%QD68	REAL
DWORD_Out		DWORD_Out	%QD69	DWORD
DINT_Out		DINT_Out	%QD70	DINT
BOOL_Out		BOOL_Out	%QX284.0	BOOL

Обращение в программе нужно осуществлять именно через объявленные переменные. Например, в программе Мастера задаются следующие значения переменным:

Expression	Application	Type	Value
MX308_EtherNet_IP_Adapter.eState	Device.Application	ADAPTERSTATE	RUNNING
IoConfig_Globals_Mapping.Byte_Out	Device.Application	BYTE	255
IoConfig_Globals_Mapping.Int16_Out	Device.Application	INT	32767
IoConfig_Globals_Mapping.Float_Out	Device.Application	REAL	1234.567
IoConfig_Globals_Mapping.DWORD_Out	Device.Application	DWORD	4022333445
IoConfig_Globals_Mapping.DINT_Out	Device.Application	DINT	1022333445
IoConfig_Globals_Mapping.BOOL_Out	Device.Application	BIT	TRUE
IoConfig_Globals_Mapping.Byte_In	Device.Application	BYTE	0
IoConfig_Globals_Mapping.Int16_In	Device.Application	INT	0
IoConfig_Globals_Mapping.Float_In	Device.Application	REAL	0
IoConfig_Globals_Mapping.DWORD_In	Device.Application	DWORD	0
IoConfig_Globals_Mapping.DINT_In	Device.Application	DINT	0
IoConfig_Globals_Mapping.BOOL_In	Device.Application	BIT	FALSE

Контроль состояния Ведомого устройства можно осуществлять через элемент структуры Device\_Name.eState  
 MX308\_EtherNet\_IP\_Adapter.eState














Данные автоматически будут попадать в регистры Ведомого устройства (Consumed Tags):

Watch 1			
Expression	Application	Type	Value
 EtherNet_IP_Module.eState	MX308_E.Application	MODULESTATE	RUNNING
 IoConfig_Globals_Mapping.Byte_In	MX308_E.Application	BYTE	255
 IoConfig_Globals_Mapping.Int16_In	MX308_E.Application	INT	32767
 IoConfig_Globals_Mapping.Float_In	MX308_E.Application	REAL	1234.567
 IoConfig_Globals_Mapping.DWORD_In	MX308_E.Application	DWORD	4022333445
 IoConfig_Globals_Mapping.DINT_In	MX308_E.Application	DINT	1022333445
 IoConfig_Globals_Mapping.BOOL_In	MX308_E.Application	BIT	TRUE
 IoConfig_Globals_Mapping.Byte_Out	MX308_E.Application	BYTE	0
 IoConfig_Globals_Mapping.Int16_Out	MX308_E.Application	INT	0
 IoConfig_Globals_Mapping.Float_Out	MX308_E.Application	REAL	0
 IoConfig_Globals_Mapping.DWORD_Out	MX308_E.Application	DWORD	0
 IoConfig_Globals_Mapping.DINT_Out	MX308_E.Application	DINT	0
 IoConfig_Globals_Mapping.BOOL_Out	MX308_E.Application	BIT	FALSE














В программе Ведомого состояние связи с Мастером можно также контролировать через элемент структуры Device\_Name.eState:

EtherNet\_IP\_Module.eState

И наоборот, если в программе Ведомого задать значения для исходящих тегов (Produced Tags):

Watch 1			
Expression	Application	Type	Value
 EtherNet_IP_Module.eState	MX308_E.Application	MODULESTATE	RUNNING
 IoConfig_Globals_Mapping.Byte_In	MX308_E.Application	BYTE	0
 IoConfig_Globals_Mapping.Int16_In	MX308_E.Application	INT	0
 IoConfig_Globals_Mapping.Float_In	MX308_E.Application	REAL	0
 IoConfig_Globals_Mapping.DWORD_In	MX308_E.Application	DWORD	0
 IoConfig_Globals_Mapping.DINT_In	MX308_E.Application	DINT	0
 IoConfig_Globals_Mapping.BOOL_In	MX308_E.Application	BIT	FALSE
 IoConfig_Globals_Mapping.Byte_Out	MX308_E.Application	BYTE	127
 IoConfig_Globals_Mapping.Int16_Out	MX308_E.Application	INT	-32768
 IoConfig_Globals_Mapping.Float_Out	MX308_E.Application	REAL	-9876.321
 IoConfig_Globals_Mapping.DWORD_Out	MX308_E.Application	DWORD	4000222333
 IoConfig_Globals_Mapping.DINT_Out	MX308_E.Application	DINT	-20123555
 IoConfig_Globals_Mapping.BOOL_Out	MX308_E.Application	BIT	TRUE

то данные автоматически попадут в регистры Мастера:

Watch 1			
Expression	Application	Type	Value
 MX308_EtherNet_IP_Adapter.eState	Device.Application	ADAPTERSTATE	RUNNING
 IoConfig_Globals_Mapping.Byte_Out	Device.Application	BYTE	0
 IoConfig_Globals_Mapping.Int16_Out	Device.Application	INT	0
 IoConfig_Globals_Mapping.Float_Out	Device.Application	REAL	0
 IoConfig_Globals_Mapping.DWORD_Out	Device.Application	DWORD	0
 IoConfig_Globals_Mapping.DINT_Out	Device.Application	DINT	0
 IoConfig_Globals_Mapping.BOOL_Out	Device.Application	BIT	FALSE
 IoConfig_Globals_Mapping.Byte_In	Device.Application	BYTE	127
 IoConfig_Globals_Mapping.Int16_In	Device.Application	INT	-32768
 IoConfig_Globals_Mapping.Float_In	Device.Application	REAL	-9876.321
 IoConfig_Globals_Mapping.DWORD_In	Device.Application	DWORD	4000222333
 IoConfig_Globals_Mapping.DINT_In	Device.Application	DINT	-20123555
 IoConfig_Globals_Mapping.BOOL_In	Device.Application	BIT	TRUE

www.OptimusDrive.ru

## Чтение и установка часов реального времени

Контроллеры семейства MX300 имеют встроенные часы реального времени (RTC). Для поддержки работы часов используется специальная модификация батарейки CR2032 (с выводом проводов), установленная в отсеке в корпусе контроллера. Для чтения и установки часов реального времени используются специальные системные команды:

- LS\_ReadDintDT
- LS\_ReadStringDT
- LS\_SetDintDT
- LS\_SetStringDT

Для их использования необходимо подключить к проекту библиотеку **MC\_SysLib**:

The screenshot shows the library manager interface. The left pane displays the contents of the selected library 'MC\_SysLib, 1.0.0.8 (Leadshine Technology Co.Ltd)', including folders like DataTypes, EtherCAT, IPAddress, LocalBus, PLCLoad, PLCReset, PLCVersion, SDCard, and SysDataAndTime. The right pane shows the 'MC\_SysLib Library Documentation' with the following details:

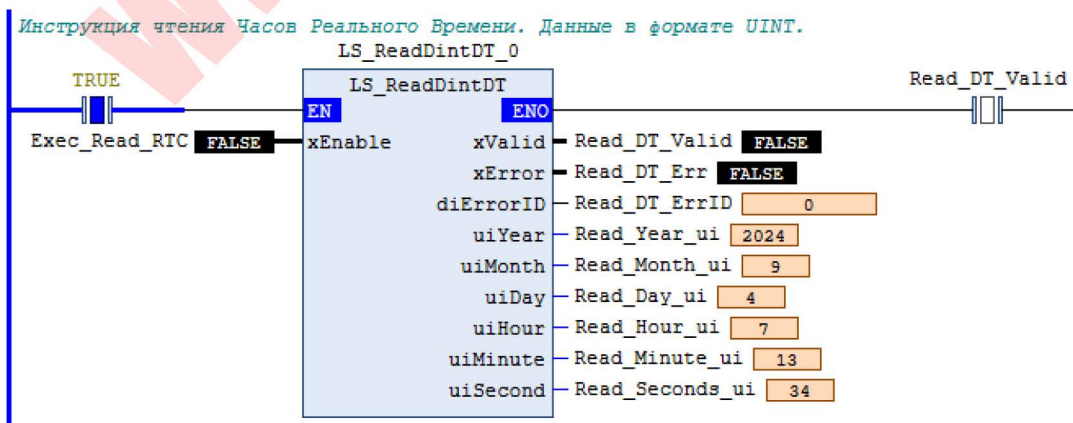
- Company: Leadshine Technology Co.Ltd
- Title: MC\_SysLib
- Version: 1.0.0.8
- Categories: LeadSys|MC
- Namespace: MC\_SysLib
- Author: Leadshine
- Placeholder: MC\_SysLib

The description section is titled 'Description [1]' and contains the text 'MC控制器专用系统功能库'.

Также, для работы наиболее употребительного функционала CODESYS к проекту должны быть подключены библиотеки **Standard** и **Util**.

The screenshot shows the library manager interface with two libraries selected: 'Standard = Standard, 3.5.18.0 (System)' and 'Util = Util, 3.5.18.0 (System)'. Both libraries are shown with their respective icons and version numbers.

Команда **LS\_ReadDintDT** читает текущее значение часов реального времени и выдаёт данные в формате UINT.





Команда **LS\_ReadStringDT** читает текущее значение часов реального времени и выдаёт данные в формате DT# (DATE\_AND\_TIME).

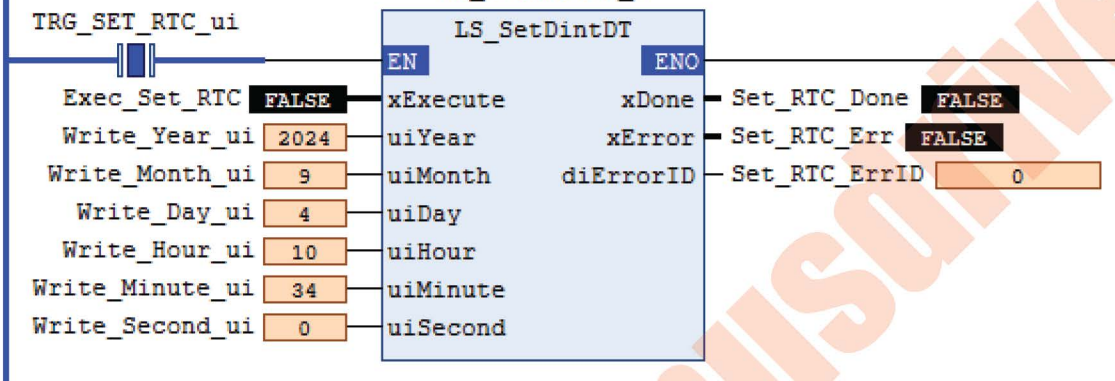
Данный тип по сути является строковым форматом типа: DT#Year-Month-Day-Hour-Minute-Second.

DT#2024-9-4-10:32:59

Для установки часов реального времени используются команды **LS\_SetDintDT** и **LS\_SetStringDT**.

Команда **LS\_SetDintDT** принимает данные в формате UINT. Данная инструкция имеет по умолчанию часовой пояс GMT + 8.

Инструкция установки Часов Реального Времени. Данные в формате UINT.  
LS\_SetDintDT\_0



Команда **LS\_SetStringDT** принимает данные в формате DT# и не учитывает часовой пояс:

Инструкция установки Часов Реального Времени. Данные в формате DT#Year-Month-Day-Hour-Minute-Second.

