



Руководство по эксплуатации

optimusdrive.ru

Оглавление

Введение.....	3
Меры предосторожности при эксплуатации	4
Функциональное назначение	5
Перечень оборудования.....	6
Спецификация ЦПУ (контроллера).....	7
Спецификация дискретных входов-выходов на ЦПУ	8
Внешний вид и размеры ЦПУ	9
Расположение клемм ЦПУ	13
Схемы подключения МН1048	14
Расположение интерфейсов МН1048	16
Спецификация источника питания.....	19
Операции со встроенным дисплеем.....	20
Запуск среды программирования и создание проекта.....	28
Установка описания устройства для контроллера МН1000.....	29
Определение версии библиотеки 3S SoftMotion (SM3).....	35
Добавление контроллера в проект	36
Организация связи контроллеров типа МН1000 и среды программирования. Загрузка программы. Онлайн режим	37
Изменение IP адреса контроллера из программы контроллера	45
Изменение IP адреса контроллера из среды программирования.....	52
Использование встроенных входов-выходов контроллера в обычном режиме	53
Поддерживаемые базовые типы данных	56
Список наиболее употребительных команд	57
Добавление в проект сервопривода.....	69
Работа с высокоскоростными счётчиками	83
Применение штурвального энкодера в качестве мастер-оси.....	91
Высокоскоростное сравнение.....	93
Работа в режиме ШИМ импульсов.....	99
Чтение и установка часов реального времени	100
Прерывание по сигналу на входе контроллера	103
Импульс при переводе в состояние RUN.....	107
Последовательная связь по протоколу Modbus RTU Master	109
Последовательная связь по протоколу Modbus RTU Slave	119
Связь по протоколу Modbus TCP Master	125
Связь по протоколу Modbus TCP Slave.....	134
Связь по протоколу Ethernet/IP Scanner (Master)	140
Связь по протоколу Ethernet/IP Adapter (Slave)	160
Замена батарейки часов реального времени	172
Обмен тегами CODESYS с панелью оператора	176
Работа с преобразователем частоты серии Optimus Drive AD800 по сети EtherCAT.....	205
Работа в режиме ШИМ импульсов.....	232
Работа со станцией удалённого ввода-вывода R2EC в сети EtherCAT.....	183
Связь по протоколу OPC UA в режиме сервера.....	190
Функция Touch Probe	216
Работа с USB disc.....	220

Введение

Настоящее Руководство описывает порядок работы с аппаратными ресурсами контроллера серии MN1000 и содержит информацию по техническим характеристикам контроллеров, а также основным процедурам работы с ними в среде программирования на базе CODESYS.

Редакция от марта 2025 года содержит следующие сведения:

- технические характеристики контроллеров
- установка конфигурационных файлов
- подключение контроллера к ПК
- работа с высокоскоростными счётчиками
- добавление сервопривода в проект
- работа по протоколу Modbus
- работа по протоколу Ethernet/IP
- установка IP адреса
- экспорт тегов CODESYS
- организация прерываний по входу
- высокоскоростное сравнение
- замена батарейки

и другие вопросы.

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

Меры предосторожности при эксплуатации

Контроллеры семейства МН1000 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

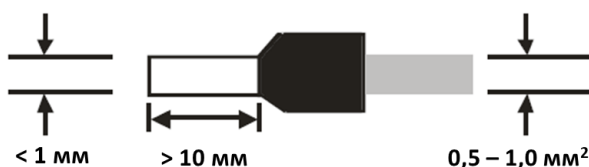
ВНИМАНИЕ

- 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 МН1000 и модулях расширения установлены пружинные клеммники, которые позволяют производить монтаж проводов без наконечников. Рекомендуются строго гибкие (многопроволочные) провода сечением 0,5 – 1,0 мм².

В случае использования наконечников рекомендуются наконечники без юбки. Если Вы принципиально используете наконечники с юбкой, то длина рабочей части наконечника должна быть 11 – 14 мм (оптимально 12 мм). Диаметр после обжима не более 1 мм.



Функциональное назначение

Приборы семейства МН1000 являются быстродействующими многофункциональными контроллерами, предназначенными для решения задач управления сложным движением по шине EtherCAT с высокой скоростью процессов и большим количеством осей движения, а также управления крупными технологическими установками и линиями. Обладают высокопроизводительным процессором Celeron и высокими программными возможностями с поддержкой сложных вычислений, логики и операций с данными. Имеют встроенные порты 2xEthernet (Gb, независимые IP адреса), 2xEtherCAT (независимые или кольцо), RS232, 2xRS485, 2xUSB-A (сохранение данных и перенос программы через flash disc). Поддерживают протоколы связи EtherCAT, Ethernet/IP, Modbus, OPC UA, socket. Минимальный такт шины EtherCAT – 250 мкс.

Для программирования контроллеров семейства МН1000 рекомендуется среда разработки Designer-AX 1.7.1 и выше версией, которую можно получить по запросу в компании Оптимус Драйв.

Возможно использование среды программирования CODESYS 3.5.18.30. 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.4.16 (4.10)
- LS Motion Lib для интерполированного движения,
- динамический E-CAM, EGear, диагностика, мониторинг данных
- Поддержка физических и виртуальных осей;
- Графический редактор E-CAM;

IP адрес по умолчанию: LAN1 (eth2): 192.168.1.3; LAN2 (eth3): 192.168.11.11

Система состоит из одного Центрального Процессорного Устройства (ЦПУ), т.е. контроллера. Модули дискретного и аналогового ввода-вывода можно использовать любых производителей через стандартные каптеры EtherCAT, для которых имеются правильные описания устройств в виде XML файлов.

Требования к системе:

Windows 10/11 64 бит

Процессор: минимум Intel Core i5 M520 2.4 ГГц

Оперативная память: минимум 8 Гб (оптимально 16 Гб)

Microsoft .Net Framework: не ниже 4.6.2 (оптимально 4.8)

Среда программирования: DIADesignerAX 1.7.1 и выше
или CODESYS 3.5 SP18 64 бит (32 бит не поддерживается)

Перечень оборудования

Наименование	Обозначение	Описание
Контроллеры (ЦПУ)	МН1048	Контроллер, 48 осей, 8DI/8DO (4 АВ счётчика 200 кГц), 2xEtherCAT, 2xEthernet(Gb), 2xUSB A, RS232, 2xRS485, 24 VDC

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 (контроллера)

Модель	МН1048
Кол-во поддерживаемых осей	48 осей EtherCAT
Процессор	Intel Celeron 4 Core 2 ГГц
Время исполнения команд	LD – 1 нс; вычисление значения float – 3 нс
Макс. кол-во модулей расширения	не расширяется модулями, необходимо использовать каплеры EtherCAT
EtherNET	2x Gb EtherNET port, Ethernet/IP, Modbus TCP, Socket, OPC UA, загрузка и выгрузка программы, отладка
EtherCAT	2xEtherCAT Master, до 64К ведомых, поддержка кольцевой топологии
Минимальный такт шины EtherCAT	250 мкс
Порт последовательной связи	RS232, 2xRS485, пользовательский протокол, MODBUS RTU Ведущий/Ведомый
RAM	4 Гб, DDR4
Hard Disc	128 Гб
Память под программу	128 Мб
Память данных	128 Мб
Энергонезависимая память	5 Мб
Память области M	5 Мб
Память области I и Q	I (%I):128 Кб, Q (%Q): 128 Кб
Порт USB	2xUSB-A 3.0, хранение данных и перенос программы на flash disc
Управление движением	Точка-точка, электронный кулачок (E-CAM), интерполяция
Высокоскоростные счетчики	4 группы АВ (200 кГц)
Встроенные входы/выходы	8 входов NPN/PNP 200 кГц 8 выходов 200 кГц, ШИМ 4 выхода OUT4 – OUT7 (NPN)
Максимальное количество точек ввода-вывода	32000
SYNC Jitter на шине EtherCAT	не более 30 мкс
Часы реального времени	Да (встроенная батарейка CR2032)
Среда разработки	Designer-AX 1.7.1, CODESYS 3.5.18.30
Языки программирования	ST, LD, CFC, SFC, FBD
Библиотеки	SM3_Basics/Robotics/CNC v.4.16 (4.10)
Напряжение питания	24 В постоянного тока (DC20.4~28.8V)
Потребляемая мощность	Номинально 20 Вт (пусковой ток > 9 А)
Рабочая температура	-5 ~ 55°C
Температура хранения	-25 ~ 65°C
Относительная влажность	10 – 95 % без образования конденсата
Вибрация	Диапазон частот: 5 – 200 Гц 5 Гц ≤ f < 9 Гц, Смещение 3.5 мм, 9 Гц ≤ f < 200 Гц, Постоянное ускорение 1g
Габаритные размеры	ВхШхГ: 165×55×140 мм

Спецификация дискретных входов-выходов на ЦПУ

Спецификация дискретных входов:

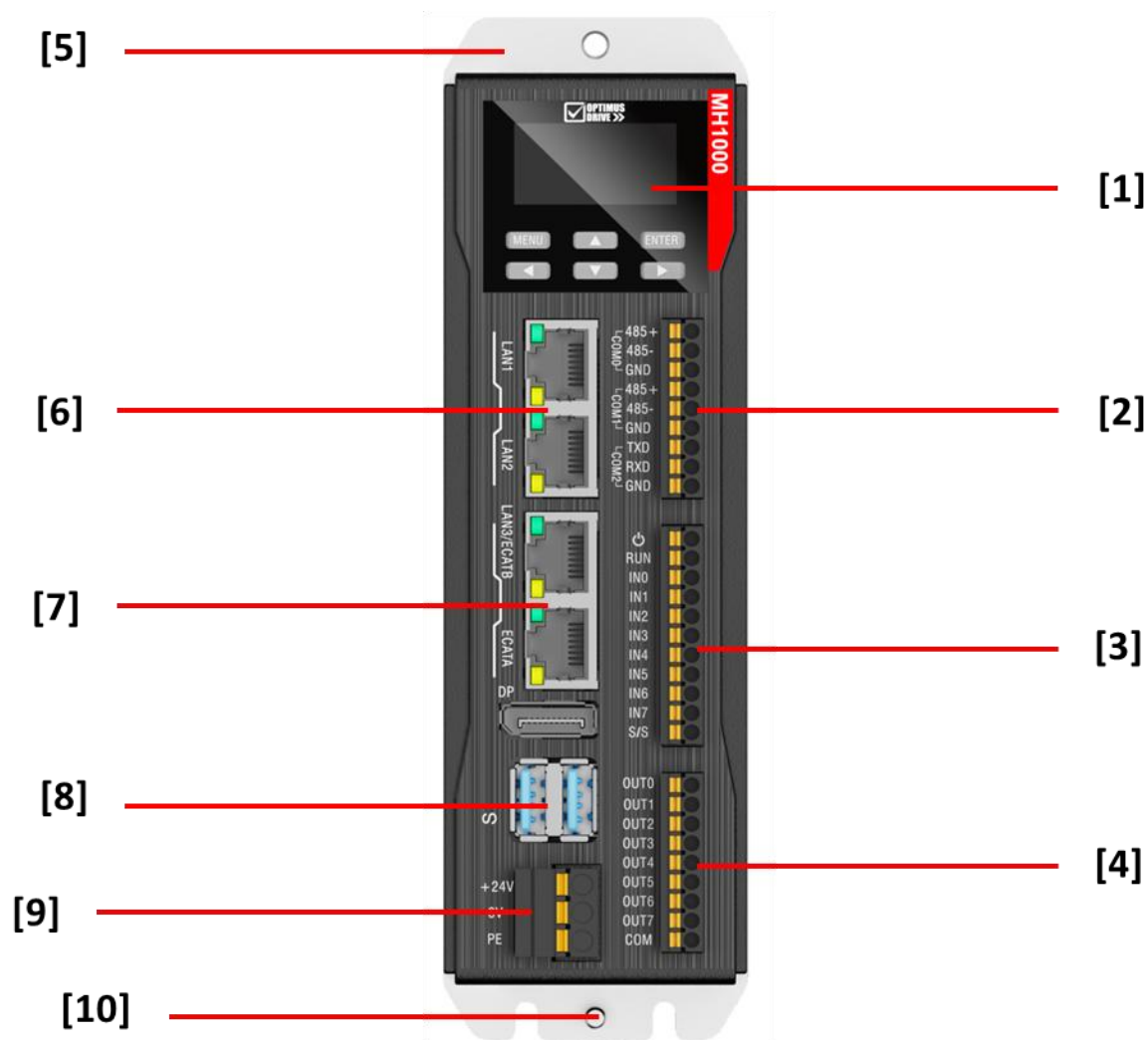
Параметр		Модель	МН1048
Количество входов			8
Тип соединения			Съёмный пружинный клеммник
Тип входа			Дискретный вход
Форма входа			Постоянный ток NPN/PNP, одна общая точка SS для IN0~IN7
Напряжение/ ток			24 VDC, 5 mA
Уровень вкл/выкл	OFF→ON		>15 VDC
	ON→OFF		<5 VDC
Максимальная входная частота			200 KHz
Входное сопротивление			4.3 kΩ
Тип входного сигнала			Потенциальный сигнал Sinking: SS подключена к 24V+ Sourcing: SS подключена к 0V
Электрическая изоляция			Оптопары, фильтр 2 мс в режиме обычного входа
Индикация			Когда оптопара активна, светодиод входа включен

Спецификация дискретных выходов:

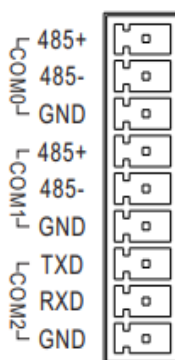
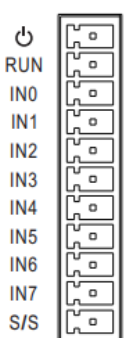
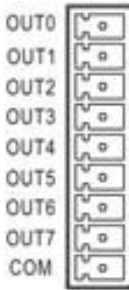
Параметр		Модель	МН1048
Количество выходов			8
Тип соединения			Съёмный пружинный клеммник
Тип выхода			NPN (Sinking)
Напряжение			5~30 VDC
Макси- мальная нагрузка	Активная		0.5A/выход, 2.4A/8 выходов на одну общую точку
	Индуктивная		-
	Лампочка		-
Максимальная выходная частота			200 KHz
Общая точка COM			COM: OUT0-OUT7

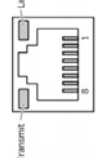
Внешний вид и размеры ЦПУ

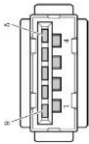
MH1048

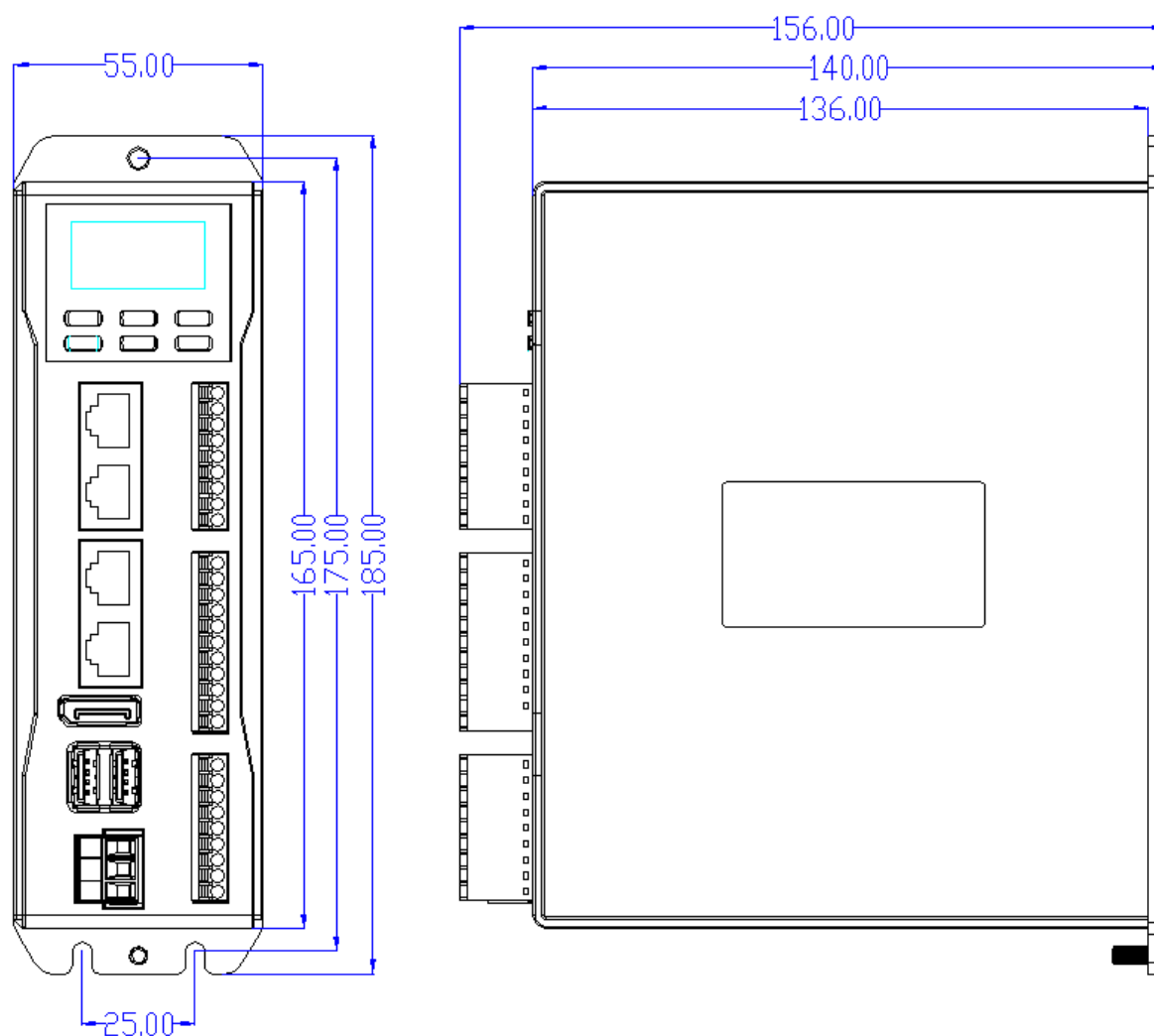


Номер	Элемент	Назначение	
[1]	Дисплей и клавиши	Дисплей отображает важную информацию и назначение клавиш следующее:	
		Клавиши	Функция
		MENU	Меню; Возврат на шаг назад
		▲	Вверх
		ENTER	Ввод
		◀	Влево
		▼	Вниз
		▶	Вправо

[2]	Последовательные порты	1 RS-232 и 2 RS-485 порта, поддержка Modbus RTU master-slave и свободного протокола Назначение клемм :				
			Пин	Сигнал	Функция	
			1	485+	1-й 485	
			2	485-		
			3	GND		
			4	485+	2-й 485	
			5	485-		
			6	GND		
			7	TXD	RS232 Tx	
			8	RXD	RS232 Rx	
9	GND	Общий для RS232				
[3]	Входы	Назначение входных клемм:				
			Номер	Обозн.	Функционал	Назначение
		1		Не используется		
[3]		2	RUN	RUN/STOP	RUN/STOP переключение ; RUN когда OFF, STOP когда ON	
		3	IN0	Вход 0	Высокоскоростные входы макс. 200 кГц, PNP/NPN	
		4	IN1	Вход 1		
		5	IN2	Вход 2		
		6	IN3	Вход 3		
		7	IN4	Вход 4		
		8	IN5	Вход 5		
		9	IN6	Вход 6		
		10	IN7	Вход 7		
		11	S/S	Общая точка входов	0 или 24 VDC	
[4]	Выходы	Назначение выходных клемм:				
			Номер	Обозн.	Функционал	Назначение
			1	OUT0	Выход 0	Высокоскоростные выходы макс. 200 кГц, NPN
			2	OUT1	Выход 1	
			3	OUT2	Выход 2	
			4	OUT3	Выход 3	
			5	OUT4	Выход 4	
			6	OUT5	Выход 5	
			7	OUT6	Выход 6	
			8	OUT7	Выход 7	
9	COM	Общая точка выходов	0V			
[5]	Шасси	Кронштейн для монтажа контроллера на панель				

[6]	Порты LAN	4 порта LAN, разъём RJ45, порядок портов сверху вниз: LAN1, LAN2, ECATB, ECATA. Индикация LAN port LED status indication :				
		Индикатор	Функционал	Цвет	Статус	Назначение
		A	A: Скорость	Зелёный/ Оранжевый	☐	OFF: 1.10Mbps скорость 2. Нет соединения
						ON_Зелёный 100Mbps скорость
						ON_Жёлтый 1000Mbps скорость
		B	B: Link/Act	Зелёный	☐	OFF: Нет соединения
						Зелёный моргает: Есть соединение идёт передача/приём данных
						ON_Зелёный Связь есть
		Описание портов:				
	LAN1	EtherNet 1	1. Default IP:192.168.1.3 2. Program download and debugging 3.ModbusTcp 4. Socket 5. EtherNet/IP			
LAN2	EtherNet 2	1. Default IP:192.168.11.11 2. Program download and debugging 3. ModbusTcp 4. Socket 5. EtherNet/I				
[7]	Сетевые порты Ethernet					
		ECATB	EtherCAT 2	1.EtherCAT 2 2.Оси и входы-выходы Сканирован ие сети		
		ECATA	EtherCAT1	1.EtherCAT 1 2.Оси и входы-выходы 3.Сканирование сети		
		Назначение контактов	1	TX+	1000M	
		2	TX-	TRD+(0)		
		3	RX+	TRD-(0)		
		4	NC	TRD+(1)		
		5	NC	TRD+(2)		
		6	RX-	TRD-(2)		
	7	NC	TRD-(1)			
8	NC	TRD+(3)				
	USB 3.0, подключение флэш U disk (FAT32, максимум 32G)					

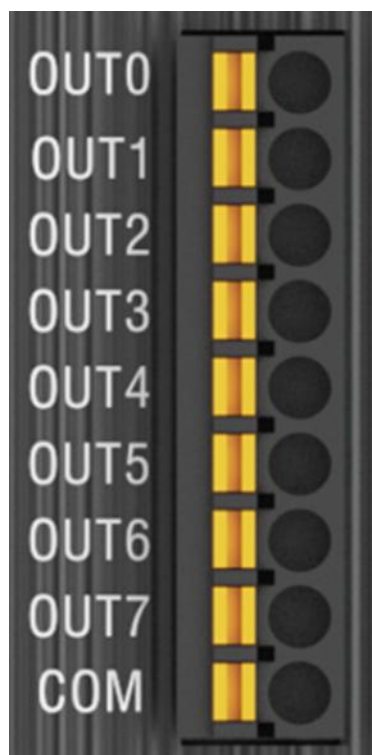
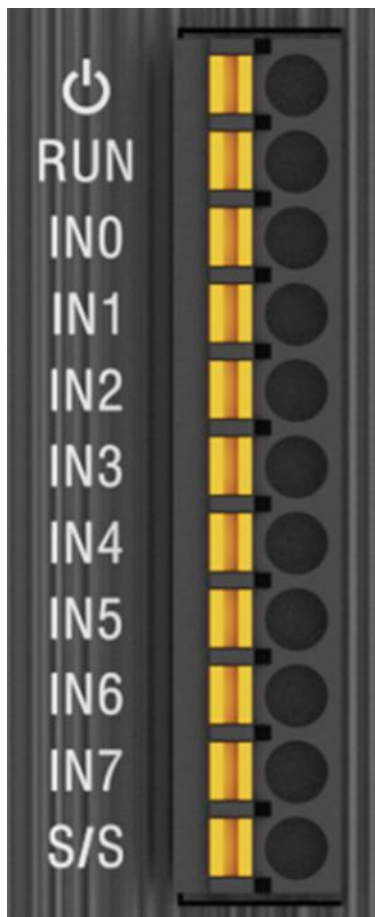
[8]	USB 1 USB 2	Назначение контактов USB 3.0:			
			Pin	Signal	Definition
			1	USB_VCC	Питание +
			2	DATA-	USB 2.0 Rx/Tx диф. сигнал
			3	DATA+	
			4	USB_GND	Питание –
			5	SSRX-	USB 3.0 Rx диф. сигнал
			6	SSRX+	
			7	USB_GND	USB 3.0 Общий
			8	SSTX-	USB 3.0 Tx диф. сигнал
9	SSTX+				
[9]	Разъём питания	+24V	Клемма питания 24 VDC		Power DC 24V
		0V	Клемма питания 0V		
		PE	Заземление		Заземление питания
[10]	Соединение	Болт заземления корпуса контроллера			



Расположение клемм ЦПУ

МН1048

Дискретные входы:



Входы IN0 – IN7 являются высокоскоростными до 200 кГц

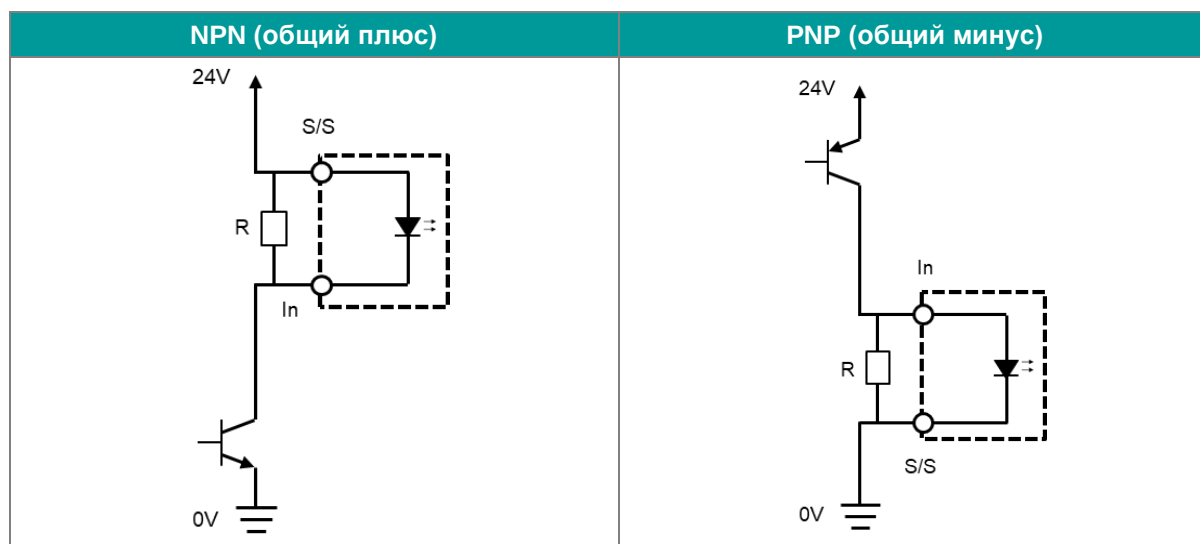
Выходы Out0 – Out7 являются высокоскоростными

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).

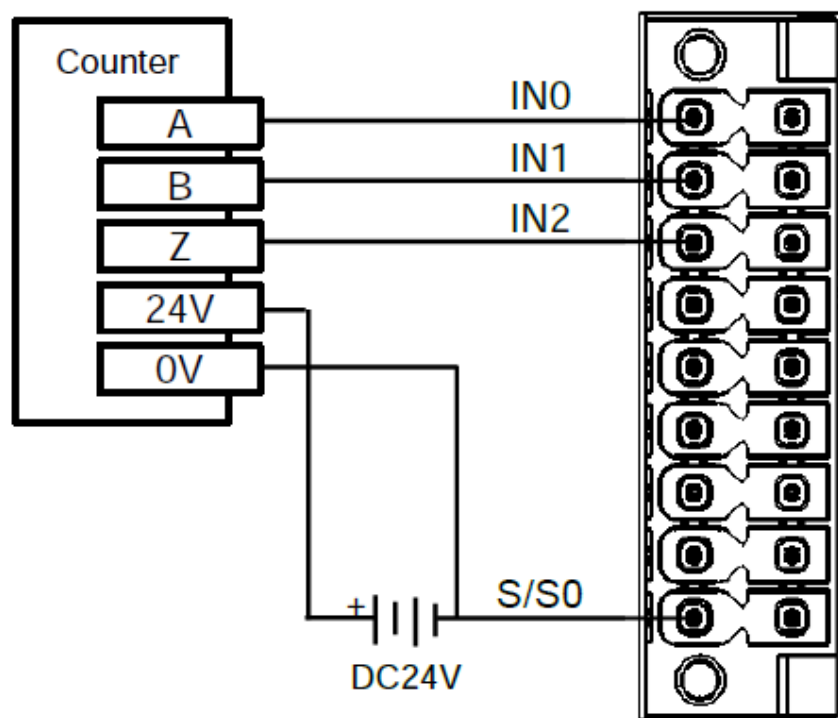
Схемы подключения МН1048

Общие рекомендации:

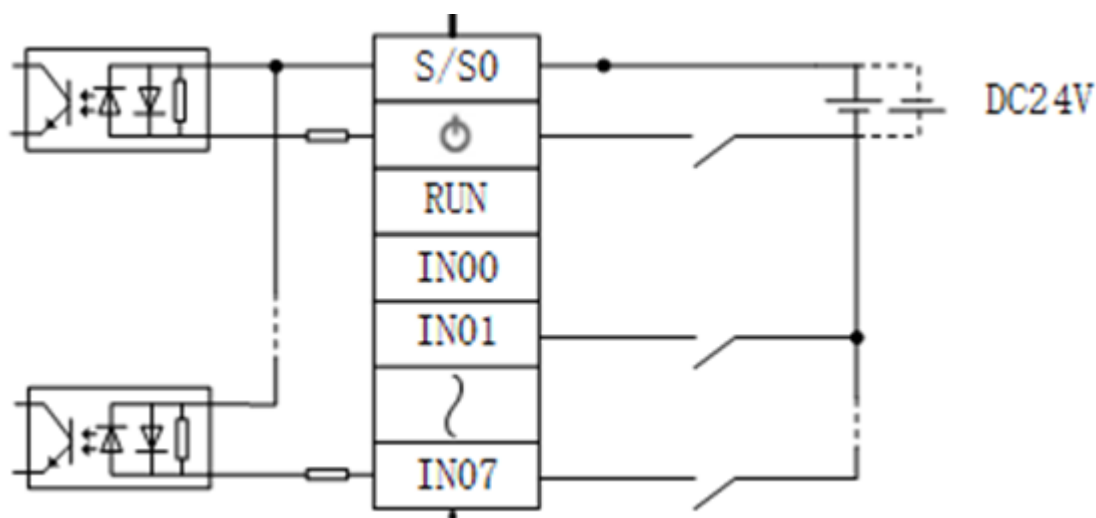
1. При подключении к контроллеру источника высокочастотного сигнала типа открытый коллектор для достижения частоты 200 кГц необходимо подключить параллельно входу и точкой S/S резистор номиналом 3 Вт/470 Ом или 2 Вт/1 кОм. На схеме ниже обозначен как R. Вход обозначен как In.
2. Если в качестве источника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



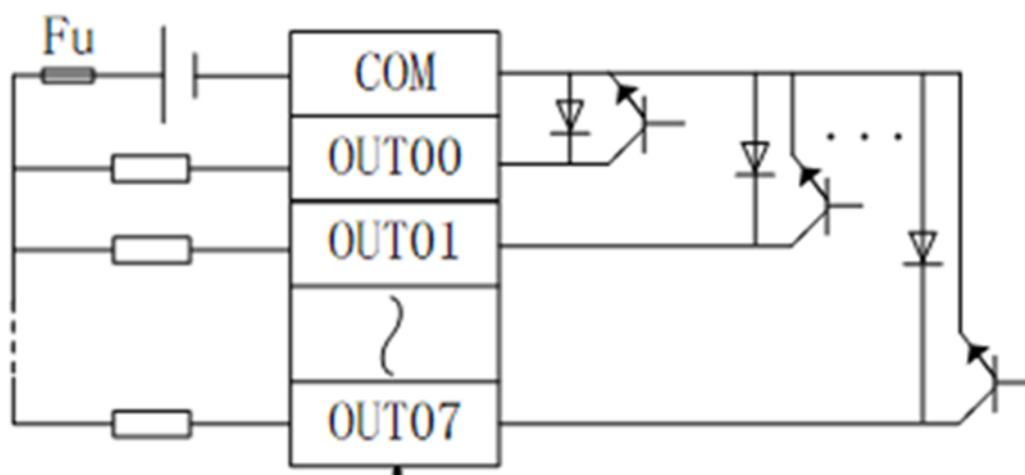
Подключение входов в режиме высокоскоростного счётчика



Подключение входов в обычном режиме



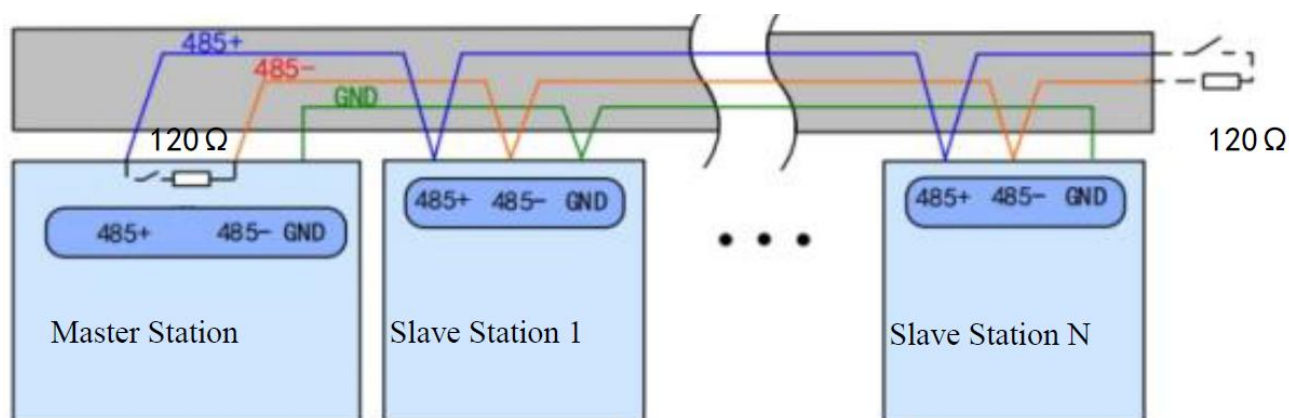
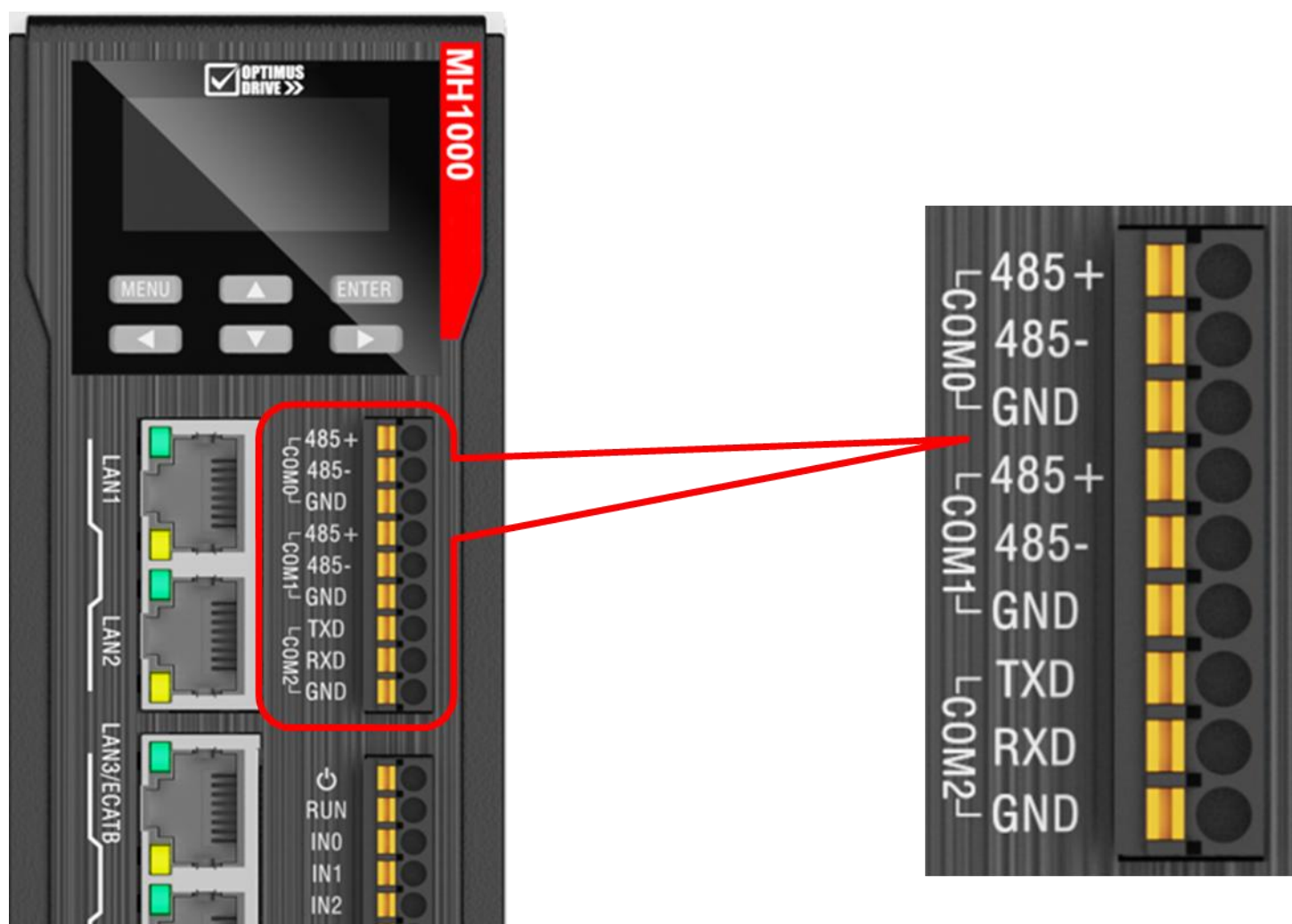
Подключение выходов типа NPN в обычном режиме



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).

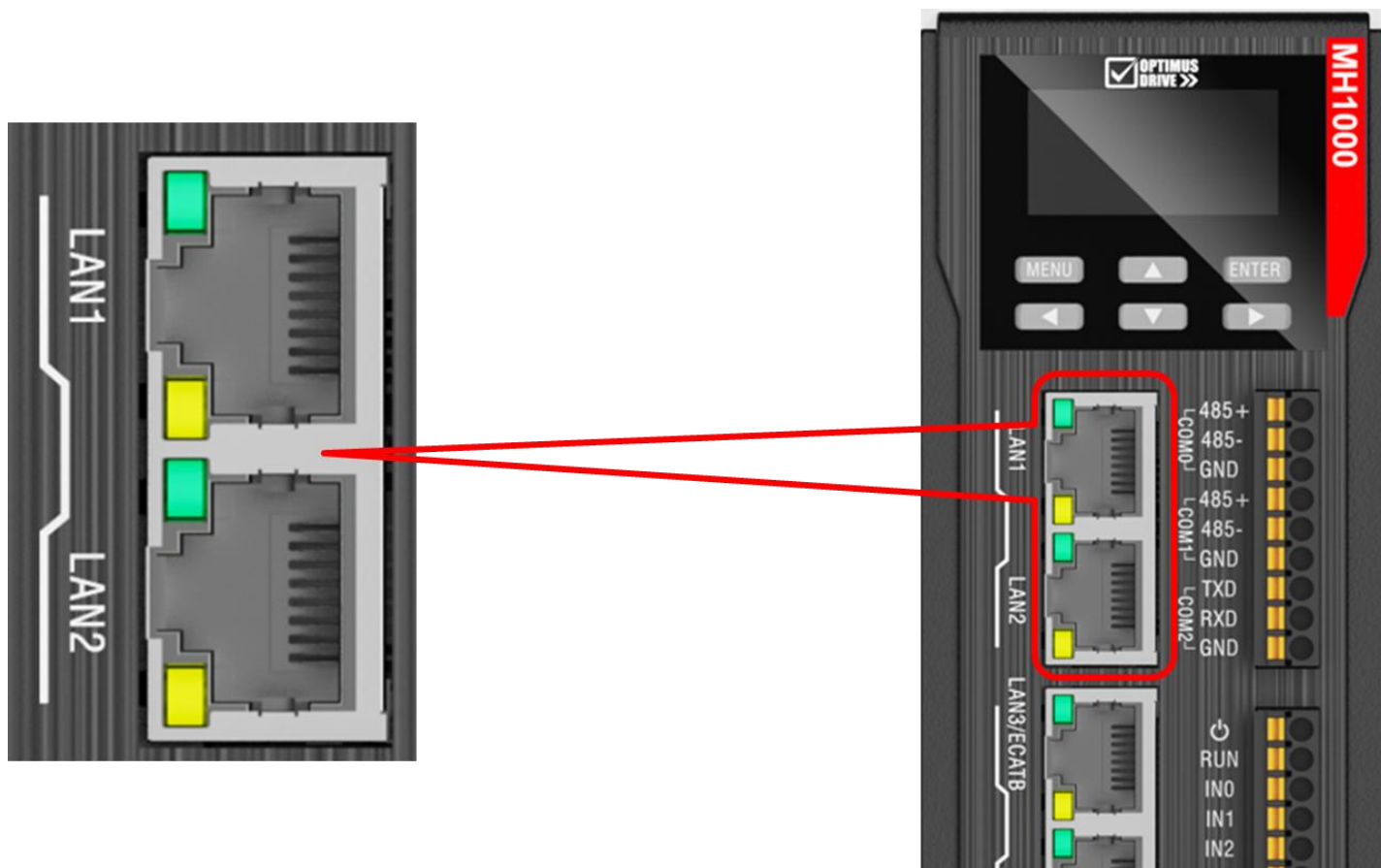
Расположение интерфейсов MH1048

RS485/RS232



Внимание! ЦПУ содержит встроенные резисторы 120 Ом для портов RS485 COM0 и COM1. Отключить их нельзя. Данный факт необходимо учитывать при проектировании сети RS485 (контроллер должен быть в начале (конце) линии).

Ethernet



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

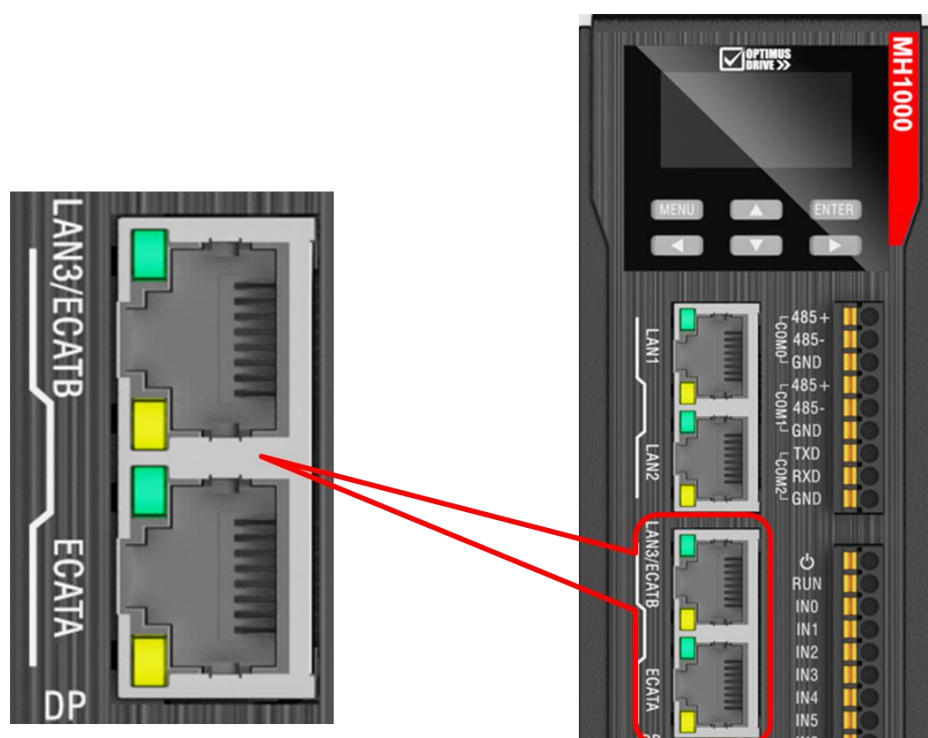
Два независимых порта, разъёмы RJ45.

IP адреса по умолчанию:

LAN1 – 192.168.1.3

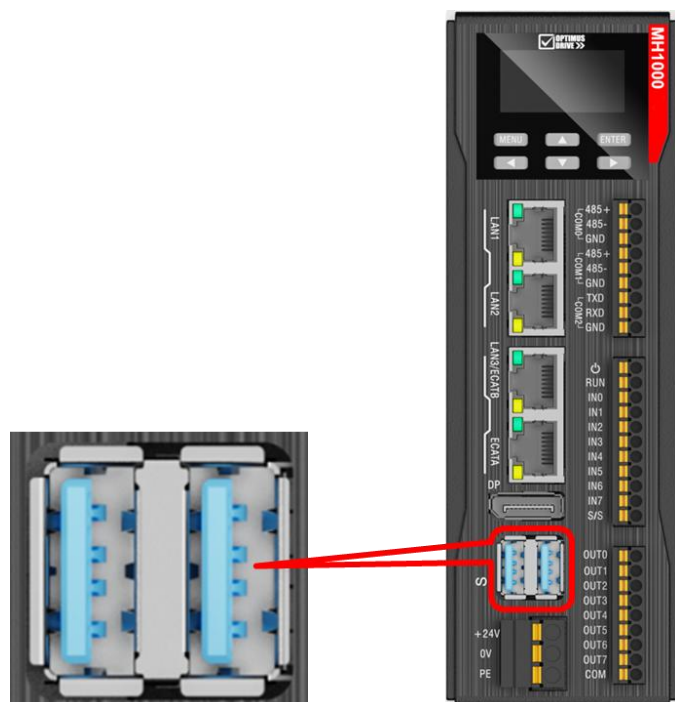
LAN2 – 192.168.11.11

EtherCAT



2 порта EtherCAT. Могут работать как независимые или в режиме резервированного кольца. Подключение необходимо выполнять пачкордами категории 6е SFTP, или специально предназначенными для сети EtherCAT

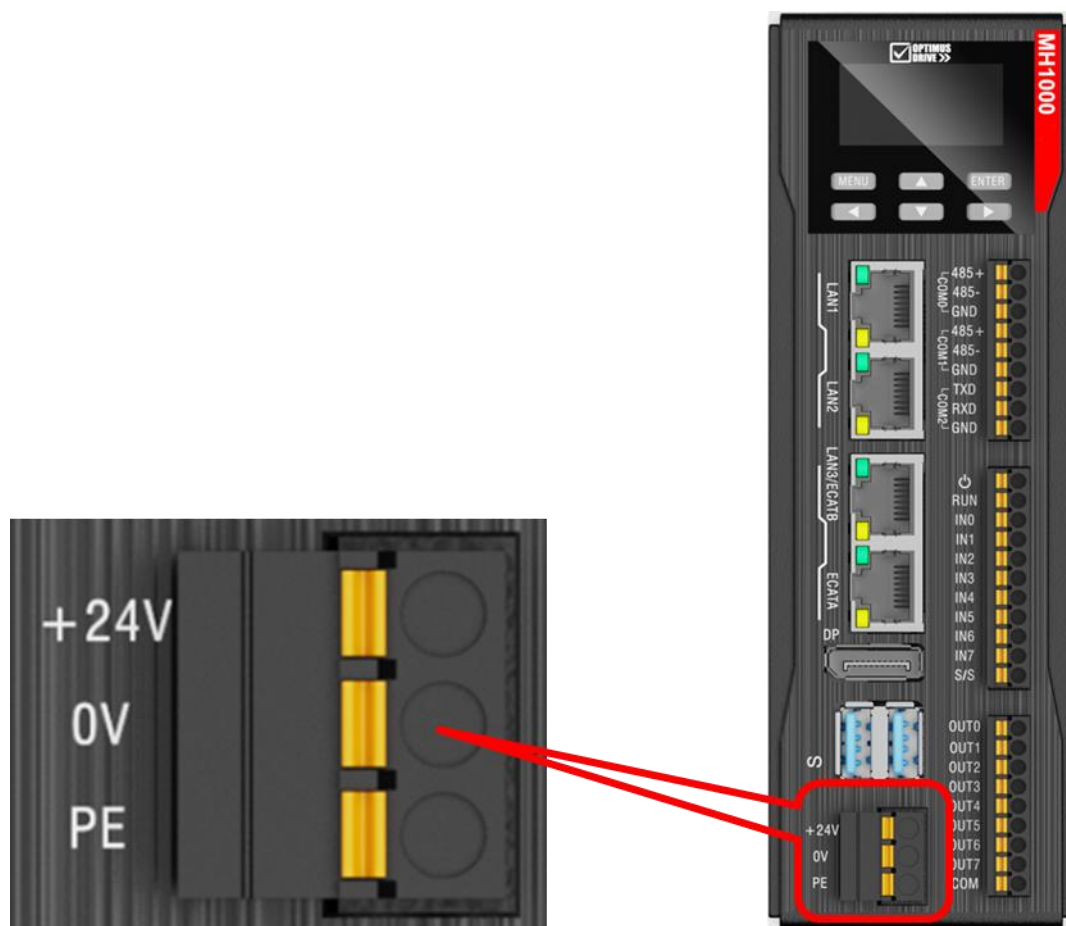
USB 3.0



2 независимых порта USB 3.0

Спецификация источника питания

Контроллеры требуют питание 24 VDC от стабилизированного источника питания. В комплекте с контроллером идёт клеммник для подключения внешнего источника питания. Рекомендованная мощность источника питания 120 Вт. Питание подключается на соответствующие клеммы:



плюс источника питания + 24 VDC подключается на клемму +24V

минус источника питания 0 VDC подключается на клемму 0V

линия защитного заземления вторичных источников питания подключается на клемму PE

Внимание! Клемму PE запрещается подключать к линии силового заземления. Линия защитного заземления вторичных источников питания должна идти отдельным проводом на вводную клемму заземления шкафа управления (линия силового заземления должна прокладываться отдельным проводом).

Операции со встроенным дисплеем

Контроллеры МН1048 оснащены жидкокристаллическим дисплеем и кнопками управления.



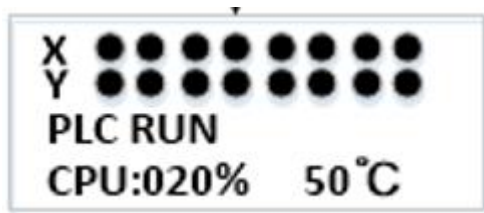
В данной Главе рассматривается работа через встроенный дисплей. Дисплей автоматически гаснет через 10 мин при отсутствии нажатий на кнопки.

Подача питания на контроллер

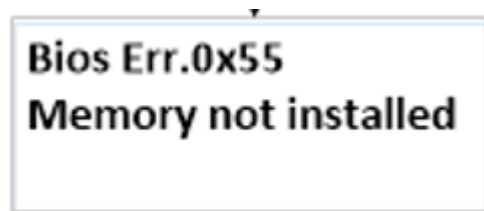
При подаче питания на дисплее будет отображаться процесс загрузки вида:

ВРС:0X32

После удачного окончания загрузки контроллера появится Главный экран, на котором отображаются состояния входов-выходов, загрузка ЦПУ, рабочее состояние RUN/STOP и температура внутри корпуса:



В случае неудачной загрузки появится сообщение об ошибке:



Список кодов ошибок приведён ниже в отдельной таблице.

Работа с меню

Нажмите кнопку **ENTER**, находясь на Главном экране. Появится меню из четырёх пунктов:



Info – Информация о системе

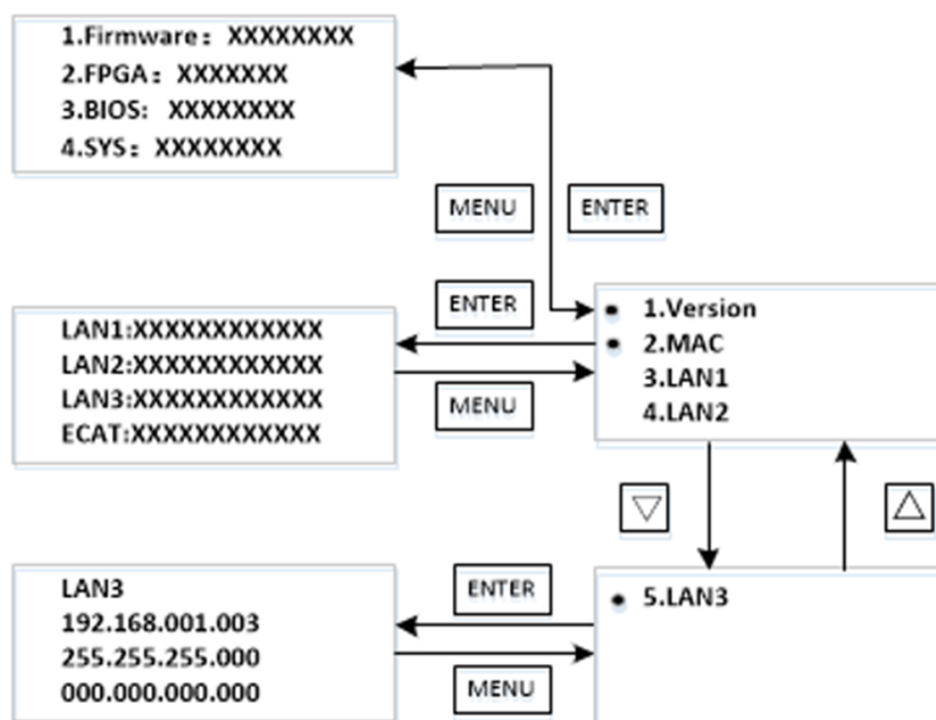
ALM – Код и текст ошибки

Monitor – Отображение загрузки ЦПУ %, памяти % и температура внутри корпуса

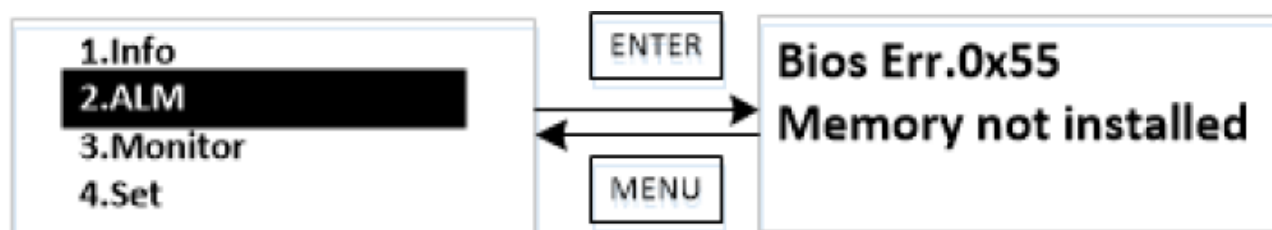
Set – Настройки: сброс IP адреса, сброс на заводские, изменение RUN/STOP, удаление программы

Для перемещения по пунктам меню используйте стрелки «**Вверх**» и «**Вниз**». Для ввода значения кнопку **ENTER**, для выхода на уровень назад кнопку **MENU**. При вводе значений пользуйтесь кнопками «**Вправо**» и «**Влево**».

Войдите в пункт **Info**. Появится отображение Версии (Version), MAC адреса и IP адресов:

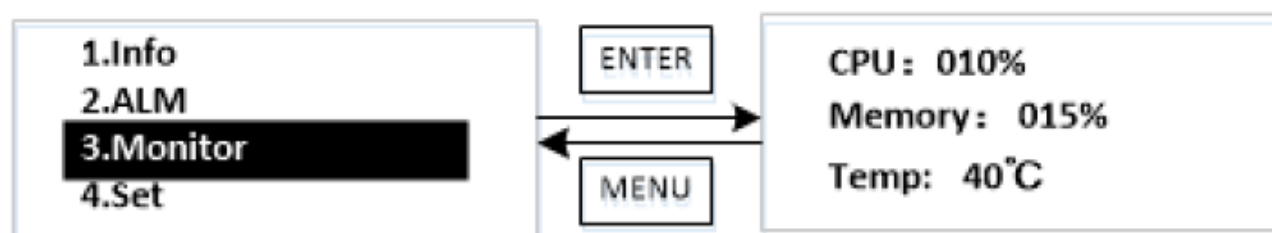


Войдите в пункт **ALM**. Появится отображение кода ошибки BIOS или кода программы:

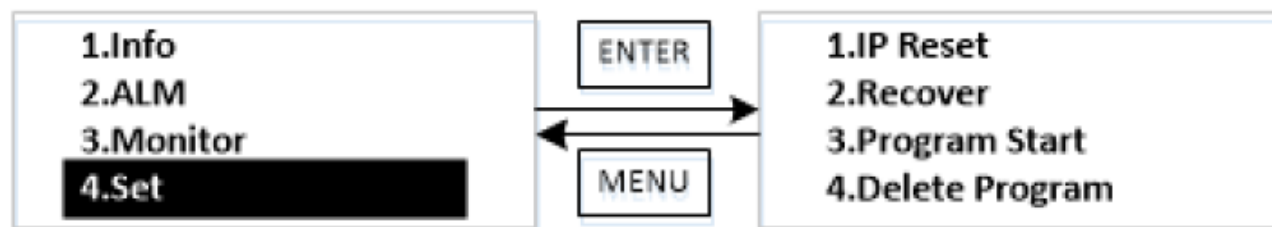


Для отображения списка ошибок используйте стрелки «Вверх» и «Вниз».

Войдите в пункт **Monitor**. Появится отображение загрузки ЦПУ %, памяти % и температура внутри корпуса:



Войдите в пункт **Set**. Появится отображение настроек: сброс IP адреса, сброса на заводские, изменение RUN/STOP, удаление программы (Application):

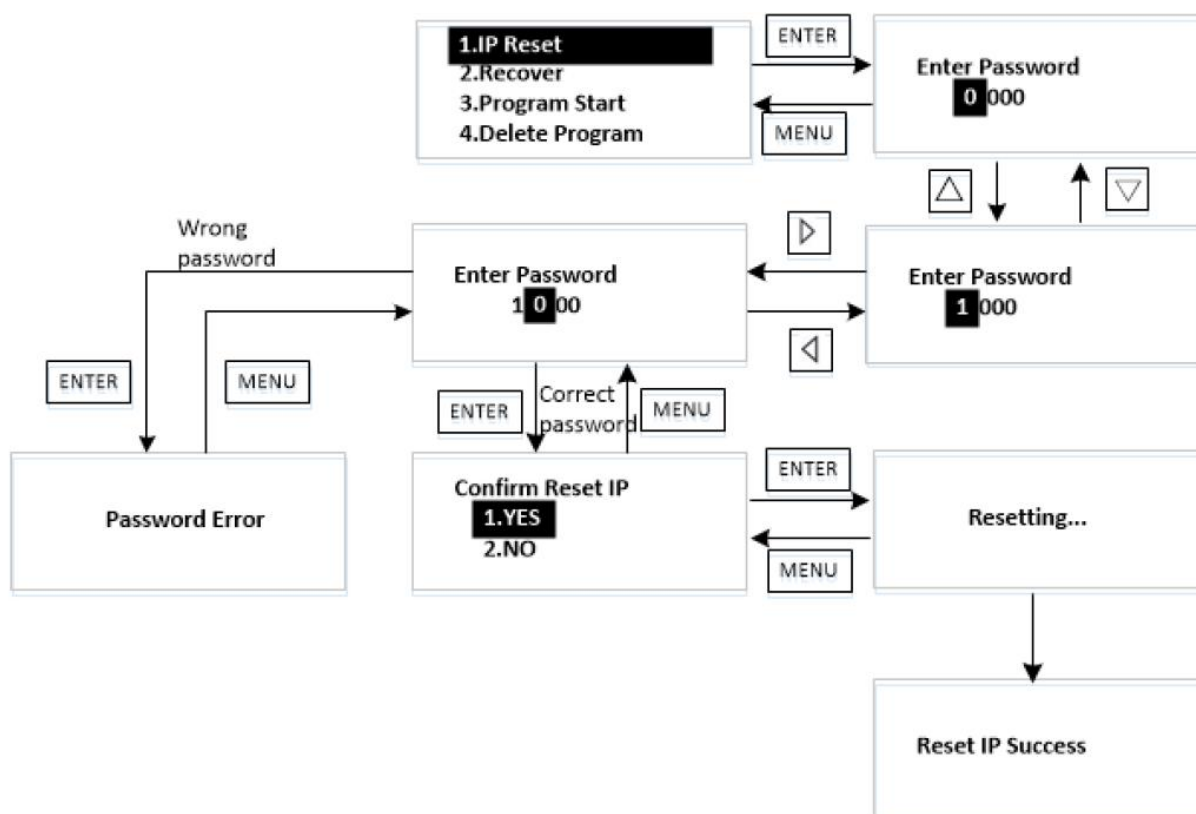


Войдите в пункт **IP Reset** для сброса адреса на заводской. Необходимо ввести пароль. Для этого используйте кнопки со стрелками. Пароль по умолчанию – 1111 (не меняется).

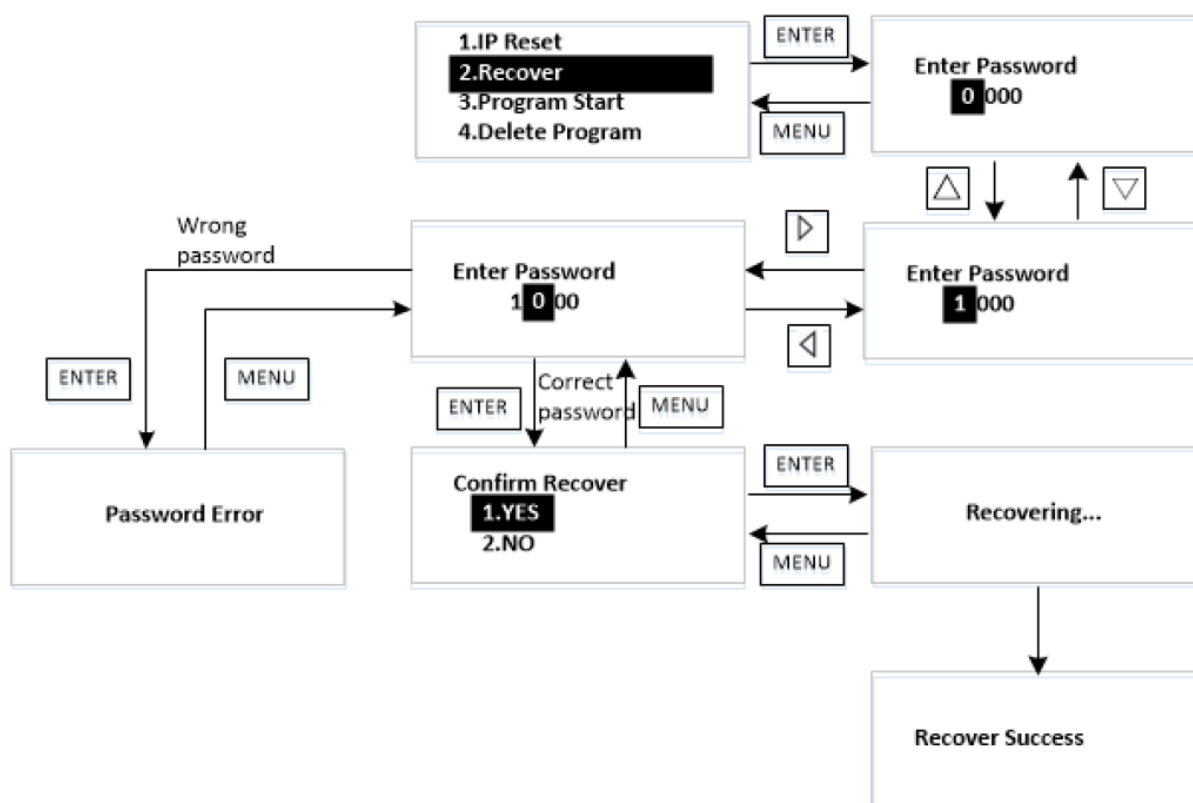
После выполнения процедуры IP адреса будут сброшены на заводские:

LAN1: 192.168.1.3

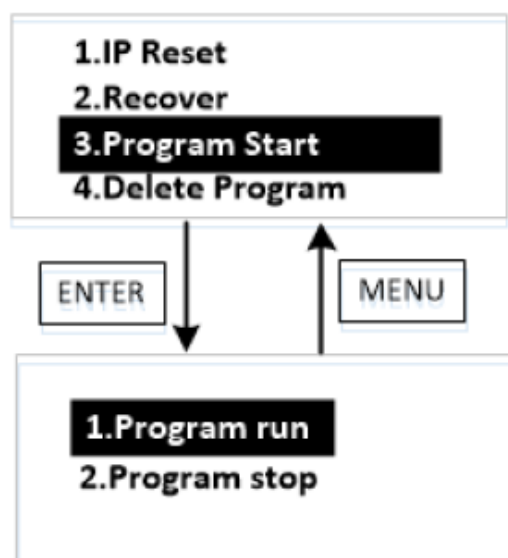
LAN2: 192.168.11.11



Войдите в пункт **Recover** для сброса на заводские настройки. Необходимо ввести пароль. Для этого используйте кнопки со стрелками. Пароль по умолчанию – 1111 (не меняется). Данная процедура аналогична команде из среды программирования Original Reset + сброс IP адресов на заводские (удаляется полностью программа, настройки и сбрасываются IP адреса).

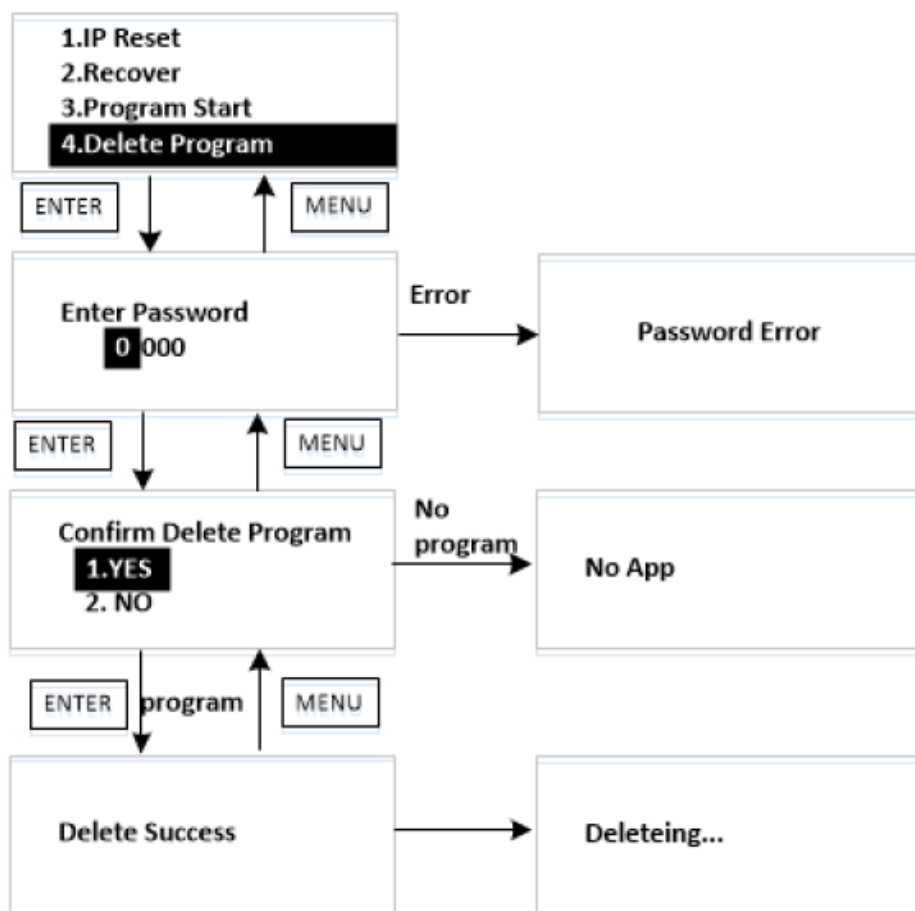


Войдите в пункт **Program Start** для изменения рабочего состояния контроллера **RUN/START**:



По умолчанию стоит режим **Program Run**. Для изменения состояния на Program Stop необходимо более 5 секунд удерживать нажатыми кнопки **MENU + ENTER**.

Войдите в пункт **Delete Program** для удаления программного приложения **Application** из памяти контроллера. Приложения с другими номерами останутся. . Необходимо ввести пароль. Для этого используйте кнопки со стрелками. Пароль по умолчанию – 1111 (не меняется).



Коды ошибок

На рабочем дисплее отображаются коды и текст ошибок BIOS и кода программы. Далее в таблице приводится их список:

Код ошибки	Описание
	NO Error
Bios Err.0x15	Pre-memory NorthBridge initialization is started
Bios Err.0x19	Pre-memory SouthBridge initialization is started
Bios Err.0x32	CPU post-memory initialization is started
Bios Err.0x3B	Post-Memory South Bridge initialization is started
Bios Err.0x4F	DXE IPL is started
Bios Err.0x60	DXE Core is started
Bios Err.0x61	NVRAM initialization
Bios Err.0x62	Installation of the South Bridge Runtime Services
Bios Err.0x69	North Bridge DXE initialization is started
Bios Err.0x70	South Bridge DXE initialization is started
Bios Err.0x72	South Bridge devices initialization
Bios Err.0x78	ACPI module initialization
Bios Err.0x79	CSM initialization
Bios Err.0x90	Boot Device Selection (BDS) phase is started
Bios Err.0x91	Driver connecting is started
Bios Err.0x92	PCI Bus initialization is started
Bios Err.0x93	PCI Bus Hot Plug Controller Initialization
Bios Err.0x94	PCI Bus Enumeration
Bios Err.0x95	PCI Bus Request Resources
Bios Err.0x96	PCI Bus Assign Resources
Bios Err.0x97	Console Output devices connect
Bios Err.0x98	Console input devices connect
Bios Err.0x99	Super IO Initialization
Bios Err.0x9A	USB initialization is started
Bios Err.0x9C	USB Detect
Bios Err.0x9D	USB Enable
Bios Err.0xA0	IDE initialization is started
Bios Err.0xA2	IDE Detect
Bios Err.0xAD	Ready To Boot event
Bios Err.0xAE	Legacy Boot event
Bios Err.0xB2	Legacy Option ROM Initialization
Bios Err.0xB4	USB hot plug
Bios Err.0x0E	Microcode not found

Bios Err.0x0F	Microcode not loaded
Bios Err.0x50	Memory init error.Invalid memory type or speed
Bios Err.0x51	Memory initialization error. SPD reading has failed
Bios Err.0x52	Memory init error.Invalid mem size or don't match
Bios Err.0x53	Memory init error.No usable memory detected
Bios Err.0x54	Unspecified memory initialization error
Bios Err.0x55	Memory not installed
Bios Err.0x56	Invalid CPU type or Speed
Bios Err.0x57	CPU mismatch
Bios Err.0x58	CPU self-test failed or possible CPU cache error
Bios Err.0x59	CPU micro-code isnot found or update is failed
Bios Err.0x5A	Internal CPU error
Bios Err.0x5B	reset PPI is not available
Bios Err.0x5C	PEI phase BMC self-test failure PEI
Bios Err.0xAB	Setup Input Wait
Bios Err.0xD0	CPU initialization error
Bios Err.0xD1	North Bridge initialization error
Bios Err.0xD2	South Bridge initialization error
Bios Err.0xD3	Some of the Architectural Protocols are not available
Bios Err.0xD4	PCI resource allocation error. Out of Resources PCI
Bios Err.0xD5	No Space for Legacy Option ROM
Bios Err.0xD6	No Console Output Devices are found
Bios Err.0xD7	No Console Input Devices are found
Bios Err.0xD8	Invalid password
Bios Err.0xD9	Error loading Boot Option (Load Imagere turned error)
Bios Err.0xDA	Boot Option is failed (StartImage returned error)
Bios Err.0xDB	Flash update is failed
Bios Err.0xDC	Reset protocol is not available
Bios Err.0xDD	DXE phase BMC self-test failure DXE
Bios Err.0xE8	S3 Resume Failed S3
Bios Err.0xE9	S3 Resume PPI not Found S3
Bios Err.0xEA	S3 Resume Boot Script Error
Bios Err.0xEB	S3 OS Wake Error
Bios Err.0xF8	Recovery PPI is not available
Bios Err.0xF9	Recovery capsule is not found
Bios Err.0xFA	Invalid recovery capsule
RT Err.0x01	Runtime crashed

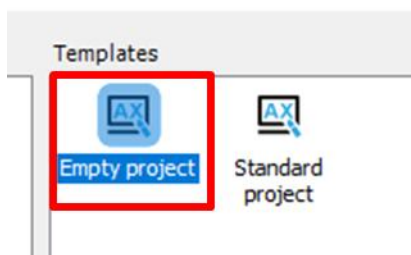
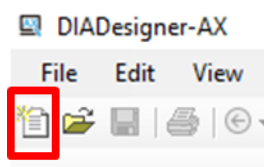
RT Err.0x02	Timeout between PLC and LCD
RT Err.0x03	I2C initialization error
RT Err.0x04	FPGA initialization error
RT Err.0x05	LSHighSpeedIO loading error
RT Err.0x06	ModbusTCP Master error
RT Err.0x07	ModbusTCP Slave error
RT Err.0x08	COM0 RS485 Modbus master loading error
RT Err.0x09	COM1 RS485 Modbus master loading error
RT Err.0x0A	COM2 RS232 Modbus master loading error
RT Err.0x7F	Battery voltage is too low
ECAT Err.0x80	EtherCAT Mastererror
ECAT Err.0x81	EtherCAT Slave error

Запуск среды программирования и создание проекта

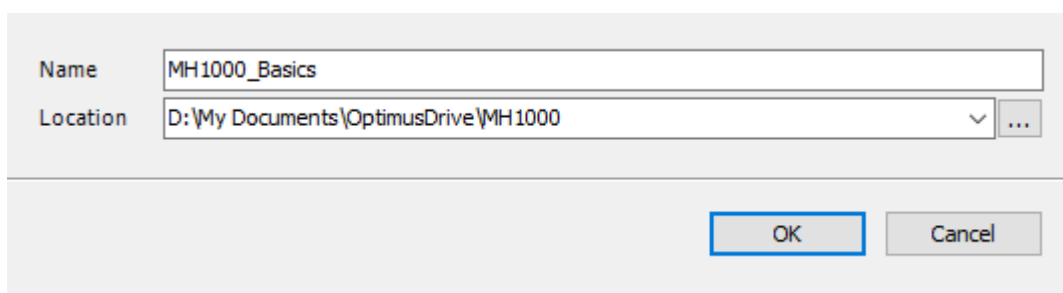
Установите на ПК (ноутбук) среду программирования DesignerAX следуя командам Мастера установки. После установки на рабочем столе появится иконка:



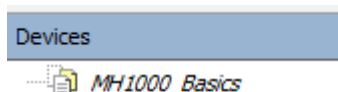
Для запуска среды программирования щёлкните дважды левой кнопкой мышки по иконке. В открывшемся рабочем окне выберите меню создания проекта и выберите создание пустого проекта «Empty Project».



Дайте название проекту:



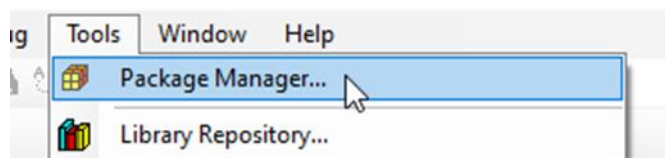
В открывшемся рабочем окне в верхнем левом углу древа проекта будет единственный пункт с названием проекта.



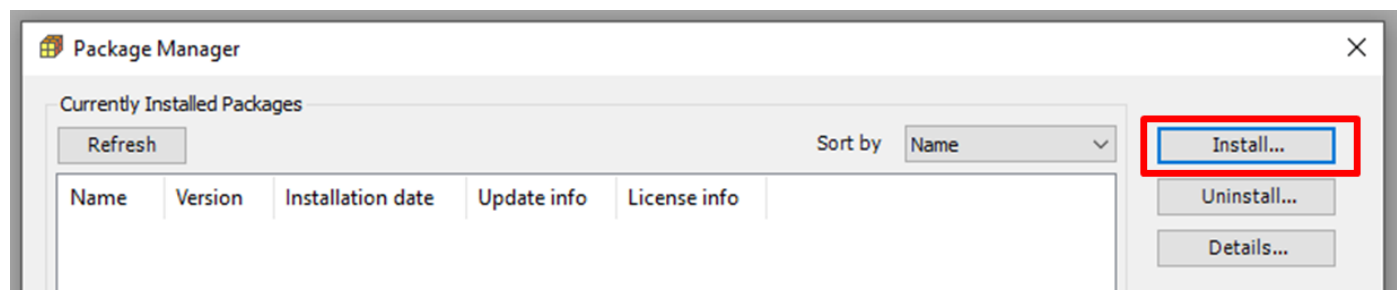
Установка описания устройства для контроллера MN1000

Для начала работы с контроллером семейства MN1000 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.

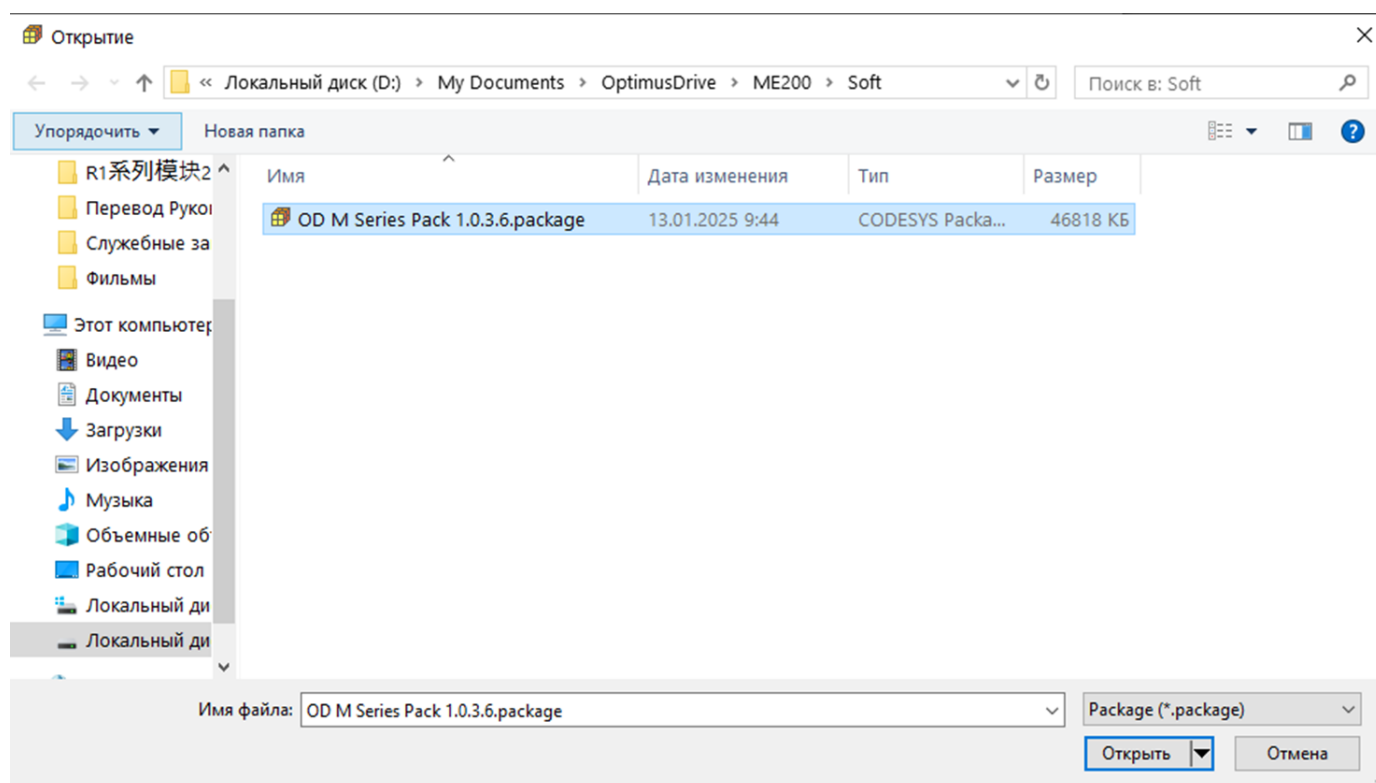
В созданном проекте войдите в меню установки пакетов устройств:



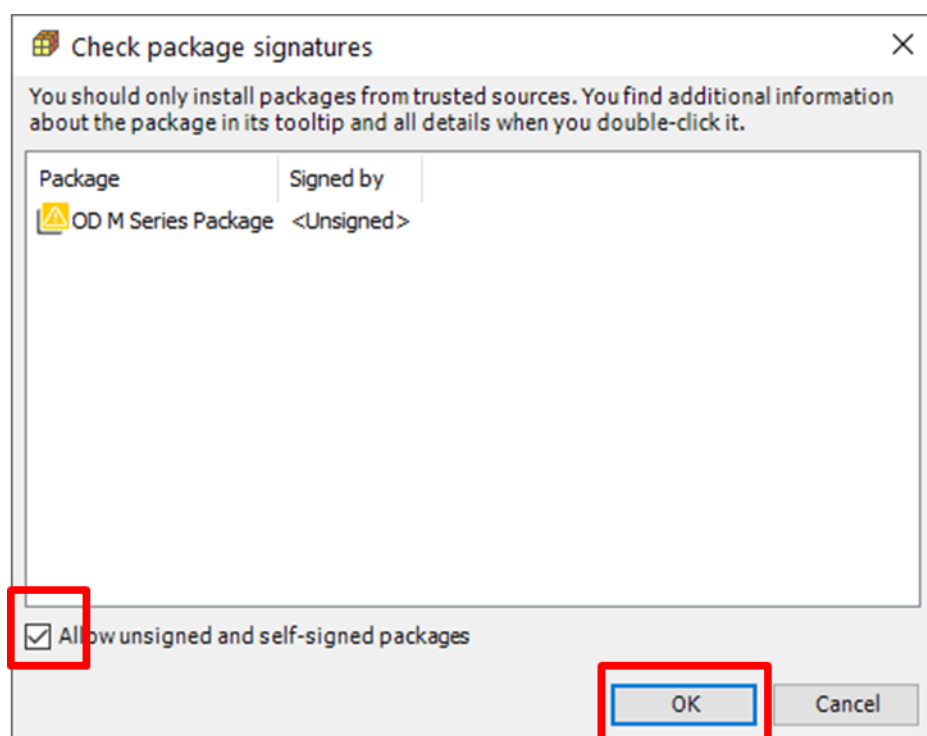
В открывшемся Мастере установки пакетов (Package Manager) нажмите кнопку **Install**:



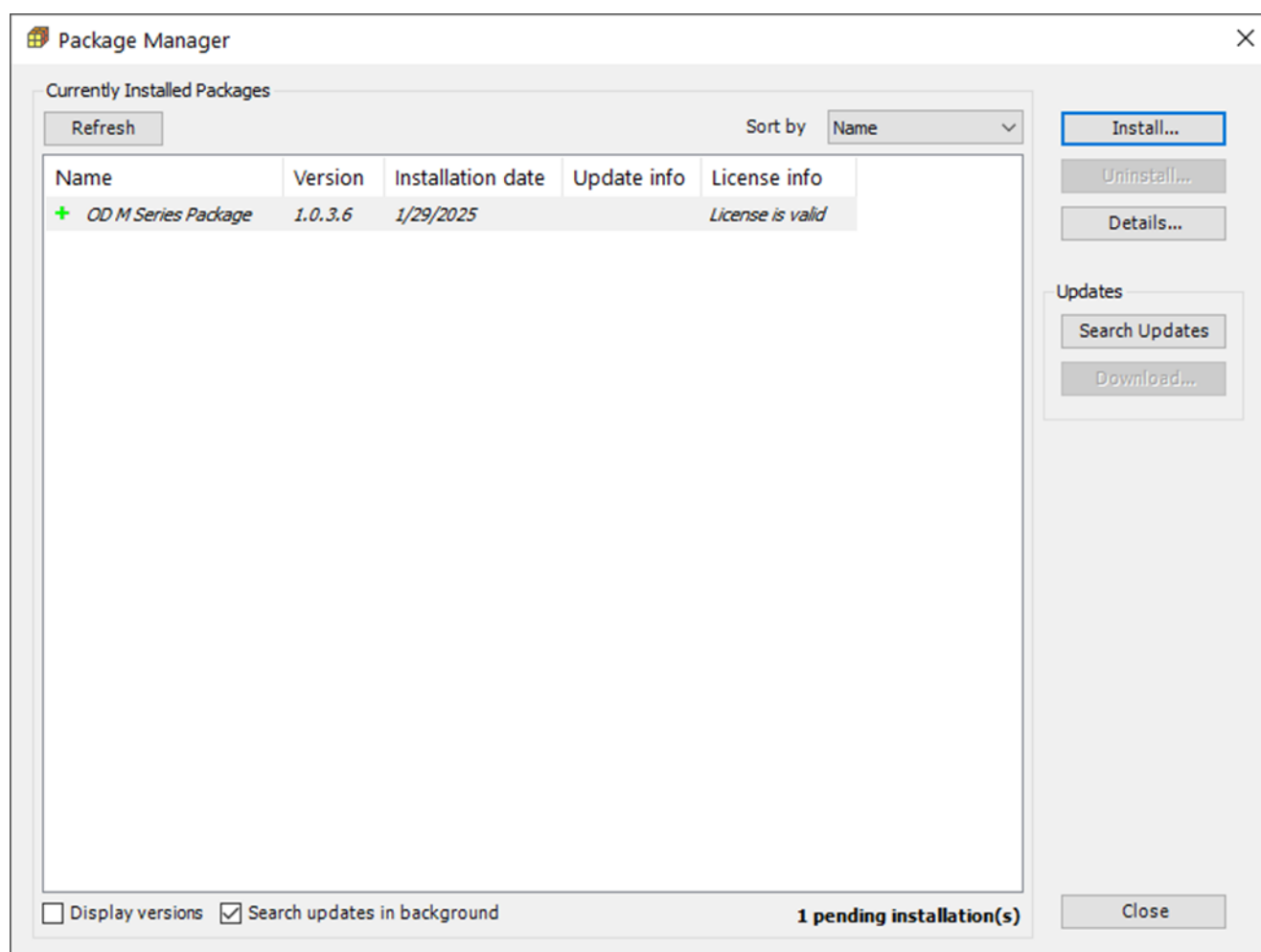
Выберите в проводнике нужный файл:



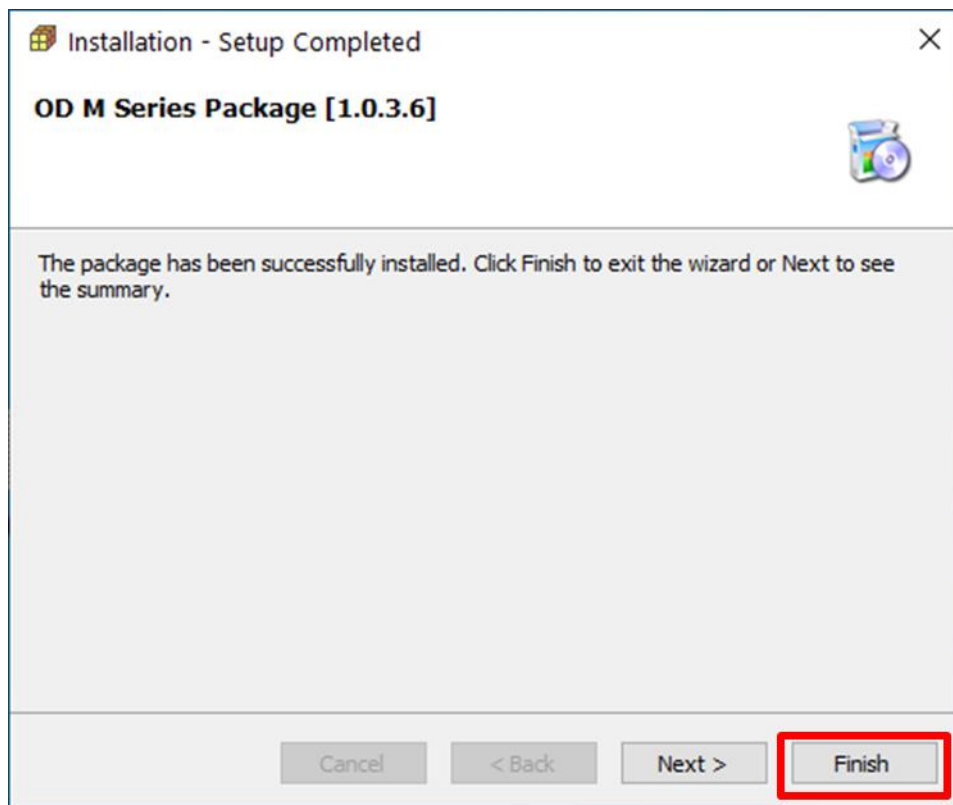
Поставьте флажок и нажмите OK:



Появится сообщение о готовности к установке пэкиджа:

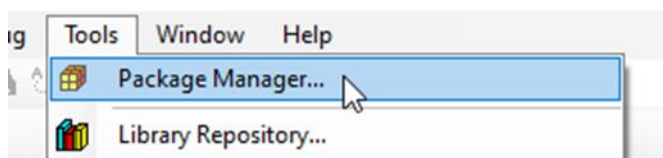


Закройте среду программирования DesignerAX, начнётся установка пэкиджа. Следуйте сообщениям на экране. По окончании установки нажмите **Finish**:

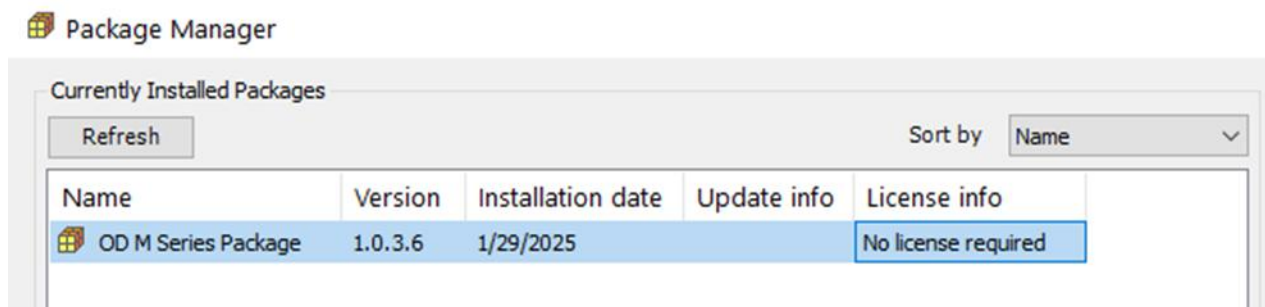


Запустите заново среду программирования DesignerAX.

Войдите в меню установки пакетов устройств:

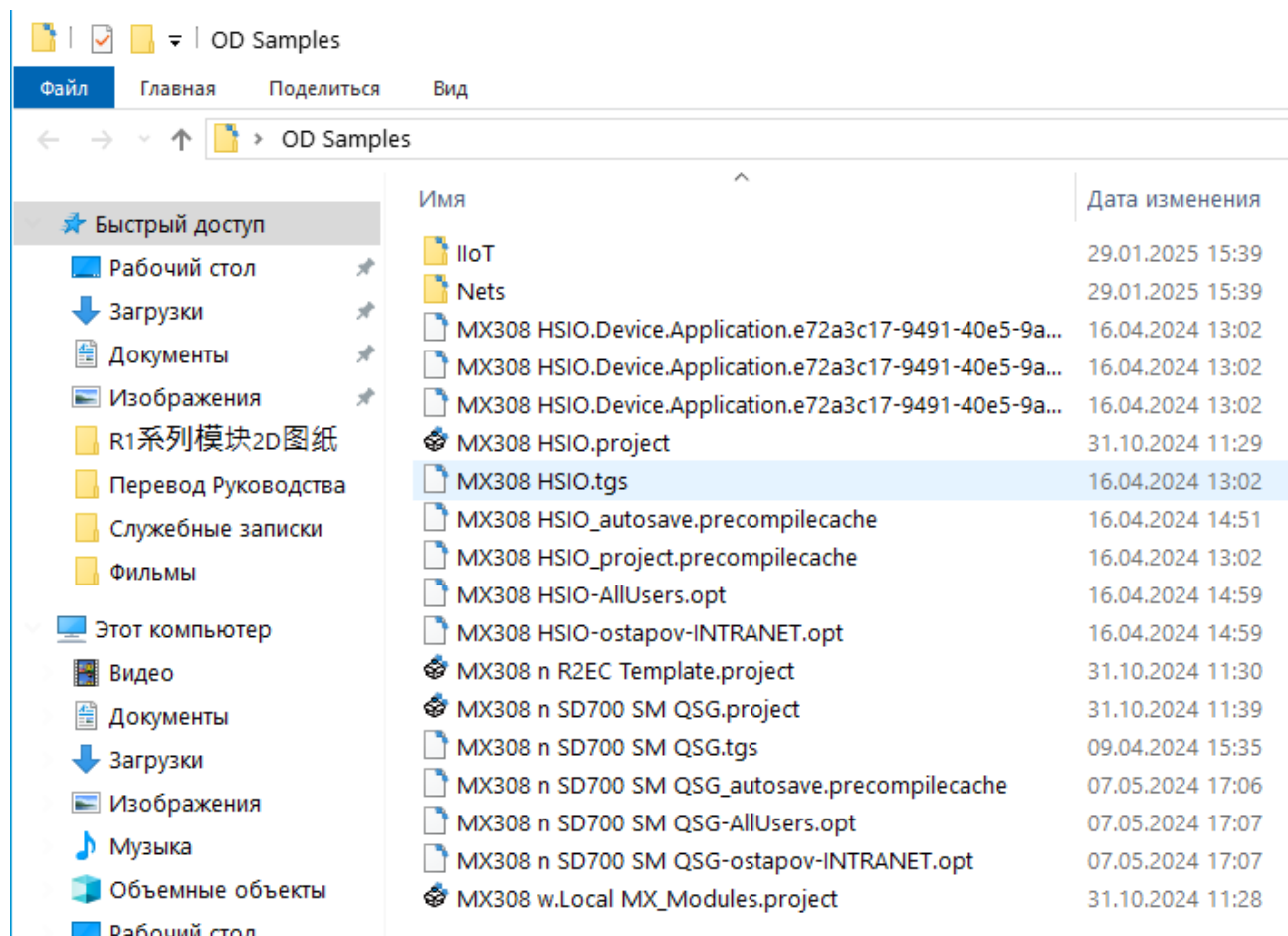
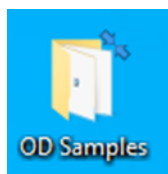


После установки появится запись в списке установленных пакетов:



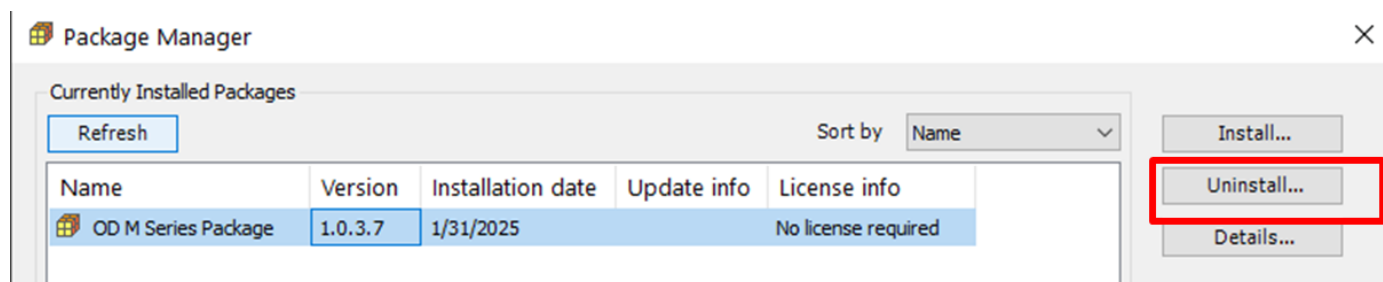
Закройте менеджер пакетов.

На рабочем столе появится папка с примерами:

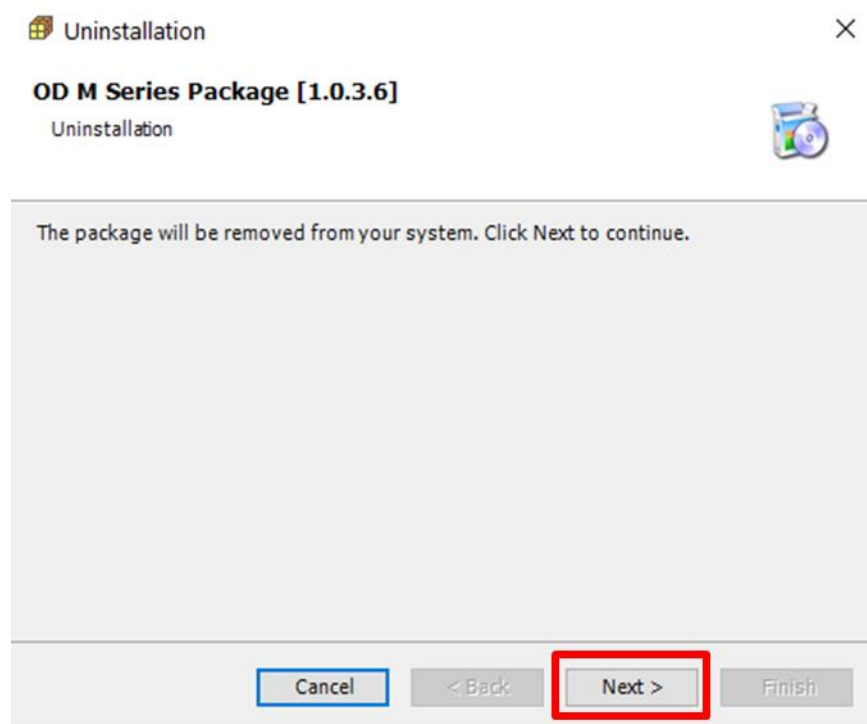


На этом установка пакета закончена и можно переходить к добавлению устройств в проект.

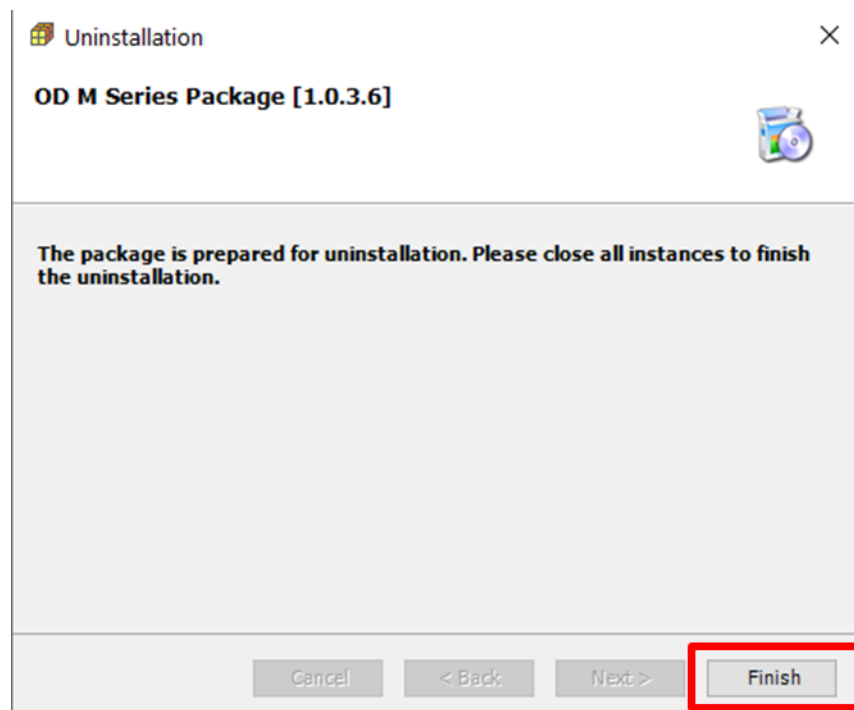
Для удаления пэкиджа необходимо нажать кнопку **Uninstall**:



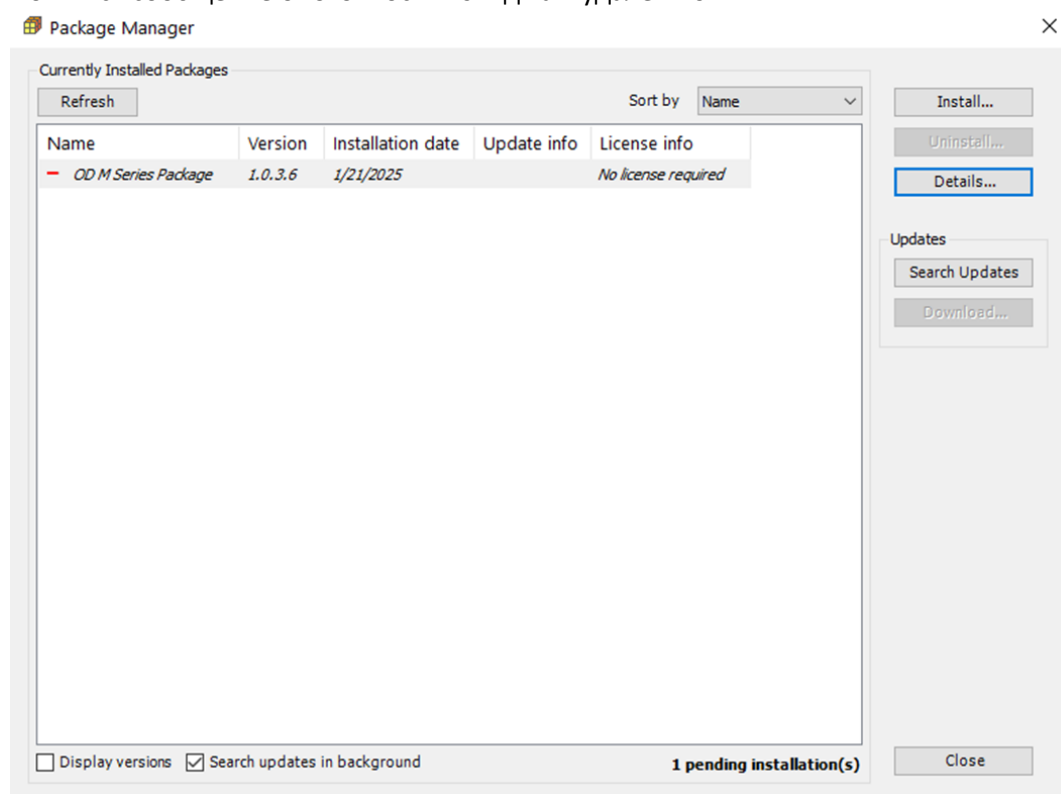
Далее **Next**:



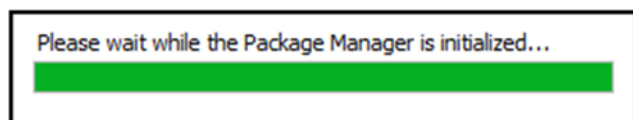
Далее **Finish**:



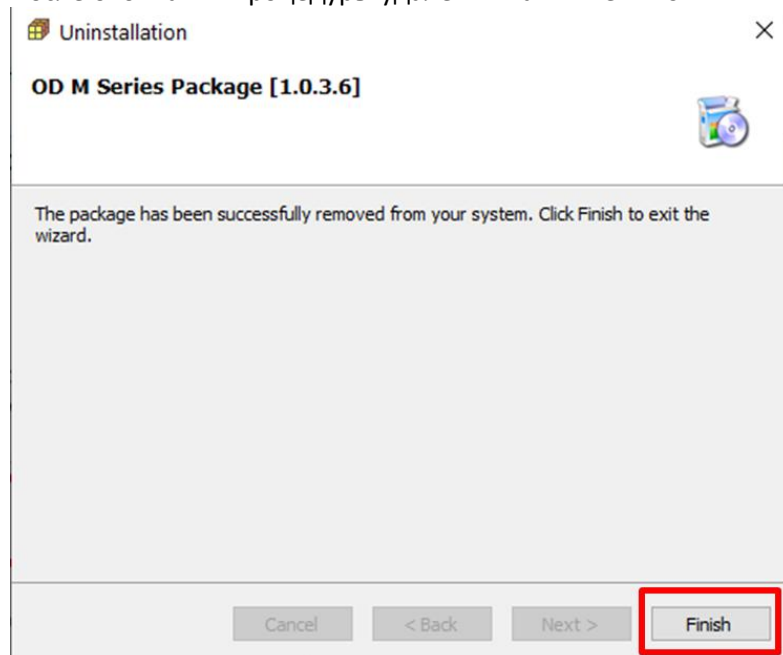
Появится сообщение о готовности пэкиджа к удалению:



После этого необходимо закрыть среду программирования. Начнётся удаление пэкиджа.



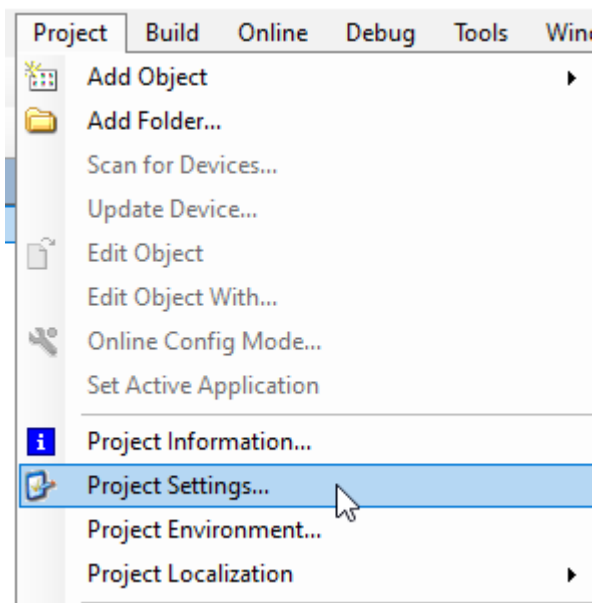
После окончания процедуры удаления нажмите **Finish**:



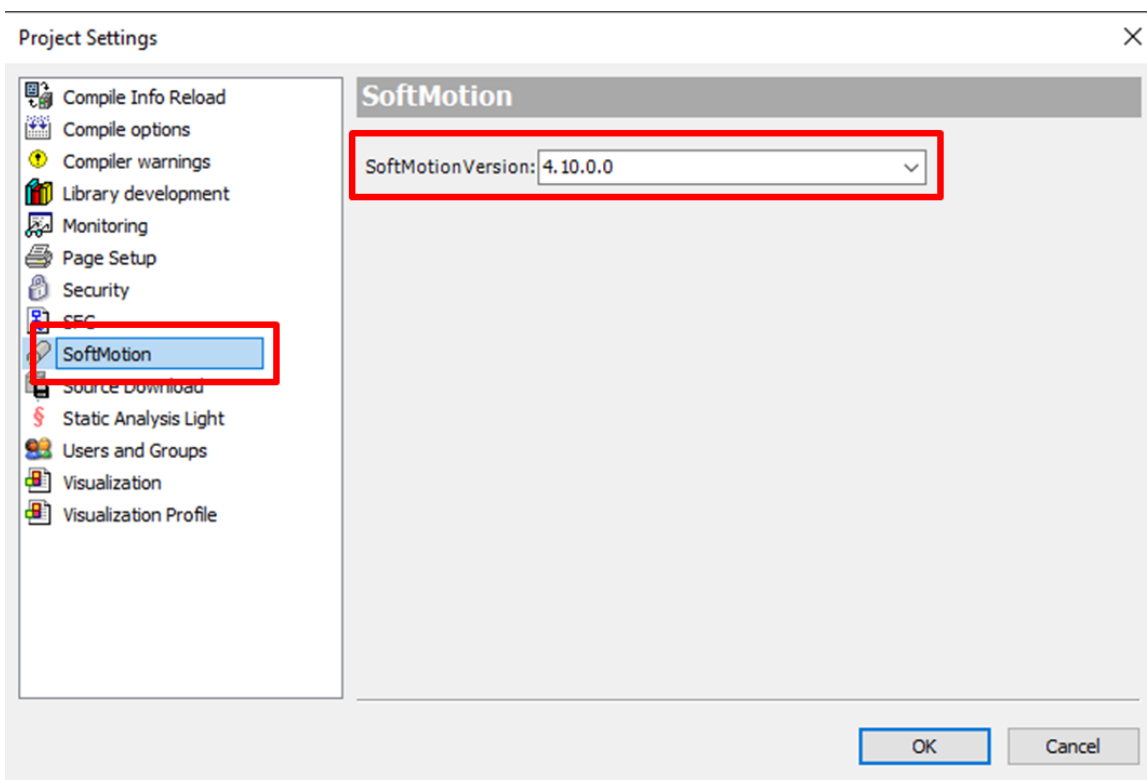
Определение версии библиотеки 3S SoftMotion (SM3)

Для версии DesignerAX 1.7 и выше наиболее подходящей версией библиотеки 3S SoftMotion является 4.16. Но можно выбрать и версию 4.10 при необходимости. Для выбора версии библиотеки 3S SoftMotion выполните следующие действия:

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

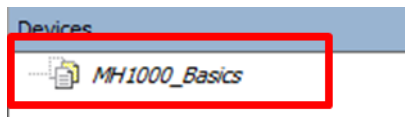


В открывшемся окне выберите пункт SoftMotion и версию 4.10.00:

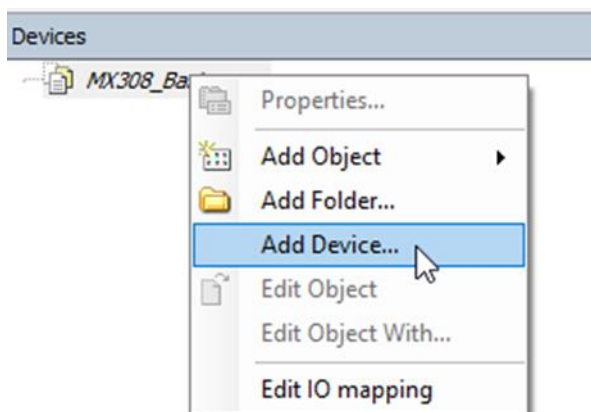


Добавление контроллера в проект

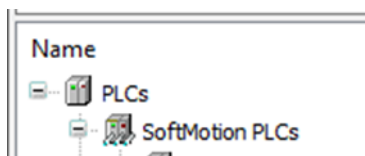
Для добавления контроллера в проект необходимо выполнить следующие действия.
Щёлкните правой кнопкой мышки на названии проекта в левом верхнем углу проекта:



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



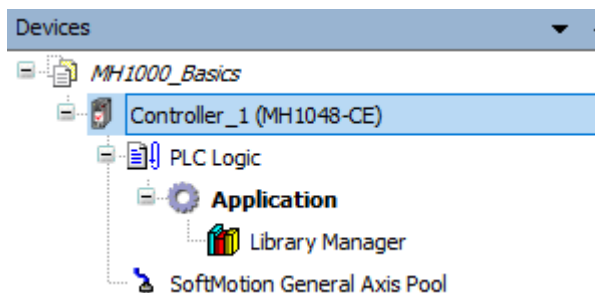
В открывшемся окне выберите раздел **PLC – SoftMotion PLC**:



Далее прокрутите вниз до пункта с контроллером MH1000

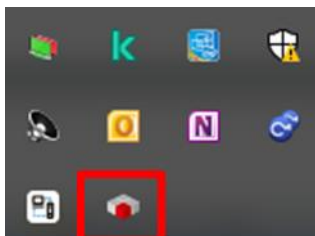
	ME204-C	Optimus Drive, Россия	3.5.15.40	Optimus Drive PLC
	MH1048-CE	Optimus Drive, Россия	3.5.15.40	Optimus Drive PLC
	MX308-CE	Optimus Drive, Россия	3.5.15.43	Optimus Drive PLC

выберите его и нажмите кнопку Add Device и закройте окно. В древе проекта появится устройство – контроллер MH1048:

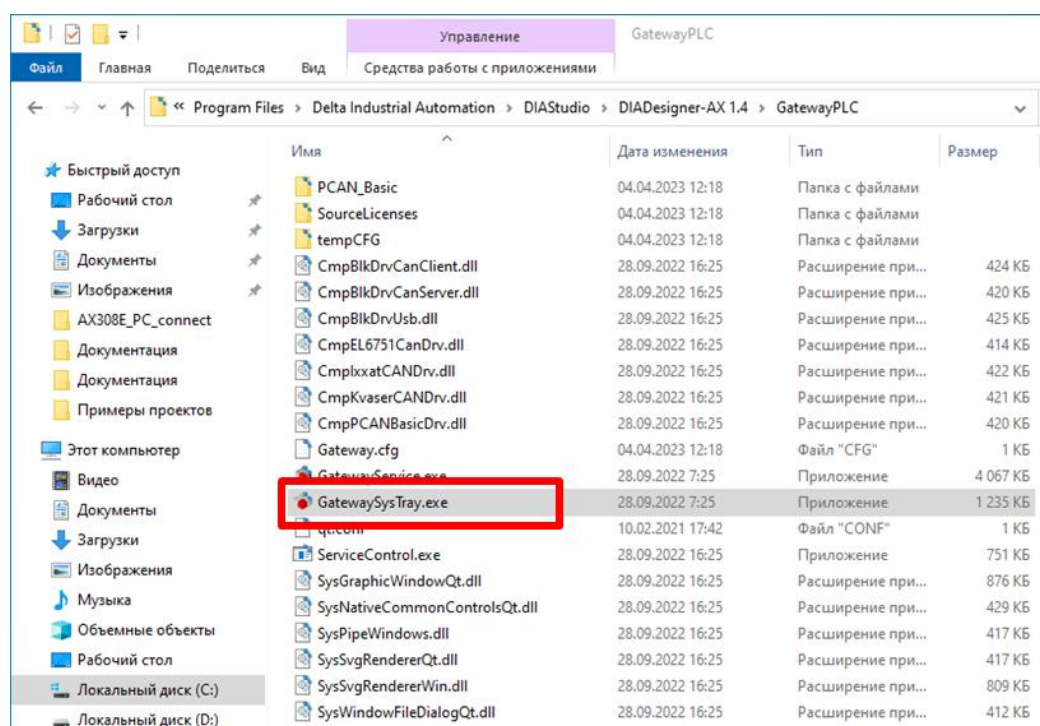


Организация связи контроллеров типа МН1000 и среды программирования. Загрузка программы. Онлайн режим

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

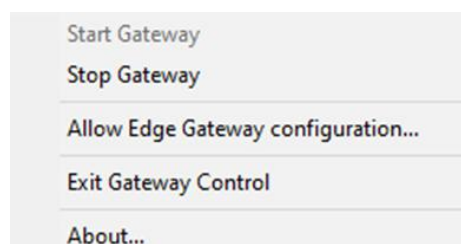


Если шлюз как приложение не запущен, то его можно запустить принудительно через исполняемый файл:



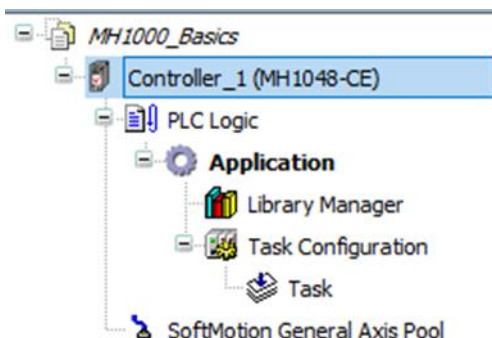
После этого должна появиться иконка.

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

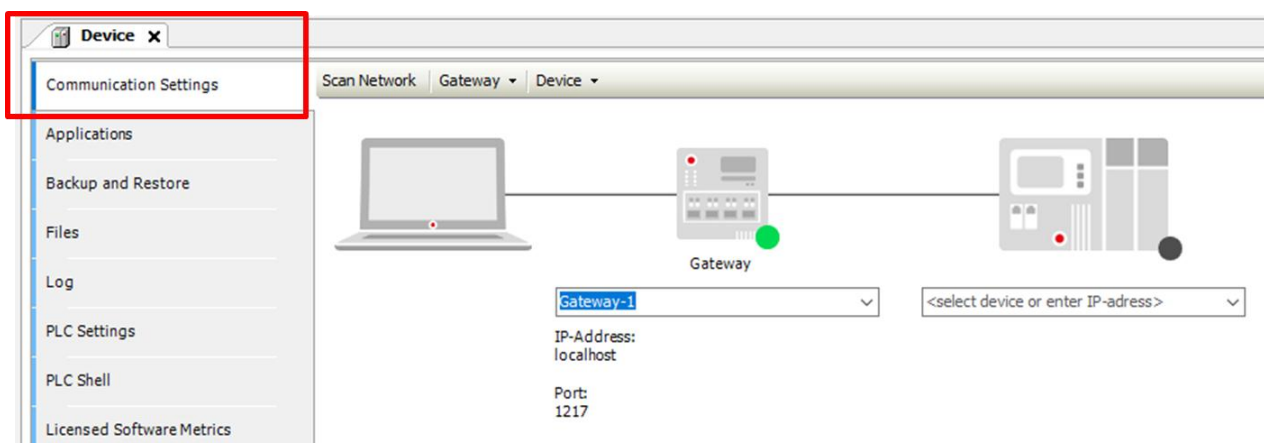


Шлюз используется для организации связи контроллера со средой программирования. Для связи можно использовать оба порта Ethernet.

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.

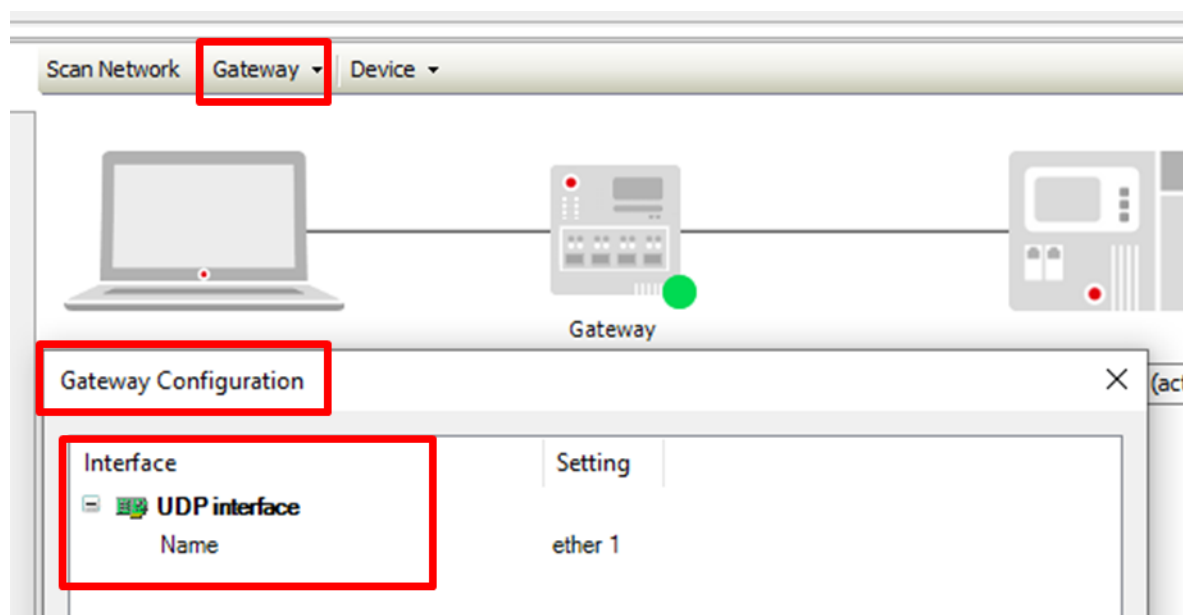


Соединение через порт Ethernet

Контроллеры серии МН1000 можно программировать через оба порта Ethernet – LAN1 и LAN2. Для установления соединения посредством сети Ethernet соедините патчкордом порты контроллера и ПК напрямую или через неуправляемый коммутатор (свитч).

У контроллера IP адрес по умолчанию: **LAN1 - 192.168.1.3, LAN2 - 192.168.11.11**

В общем случае для соединения достаточно драйвера по умолчанию:

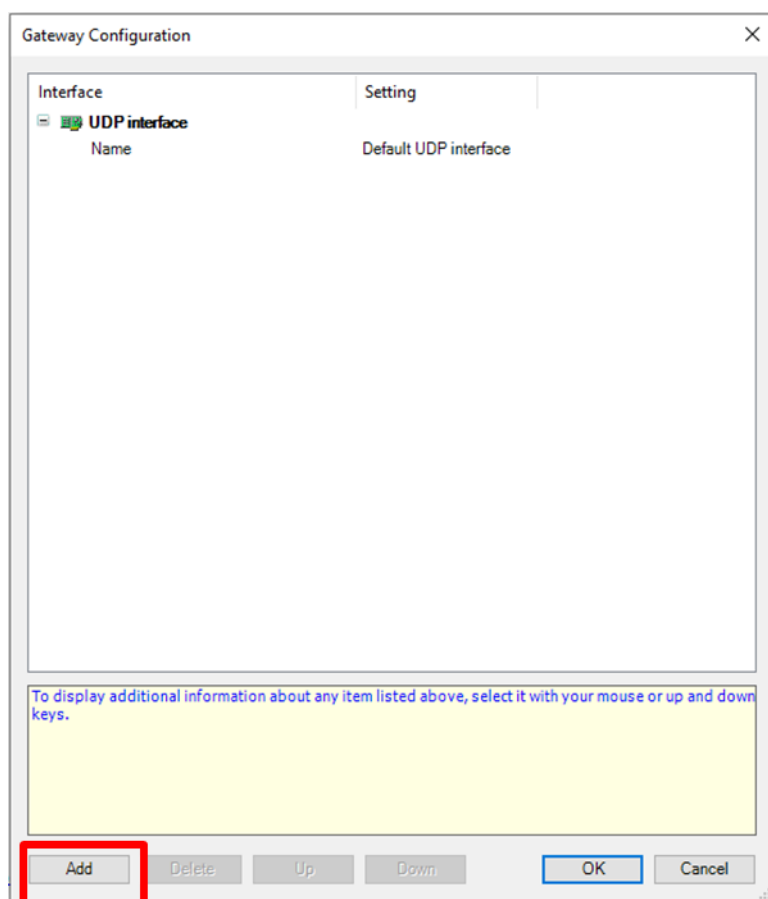


В случае соединения через несколько коммутаторов возможно потребуется создать явное TCP соединение.

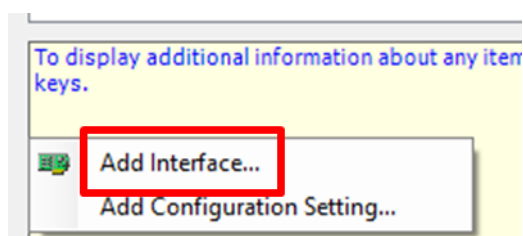
Для этого выберите пункт **Gateway – Configure the Local Gateway:**



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

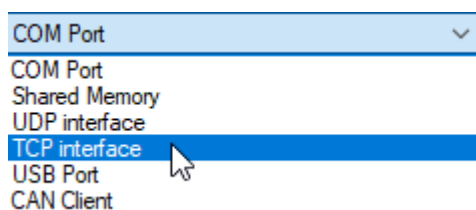


В левом нижнем углу нажмите кнопку **Add**, появится окно создания интерфейса:

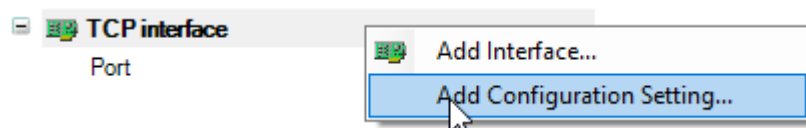


Выберите пункт **Add Interface**.

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



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

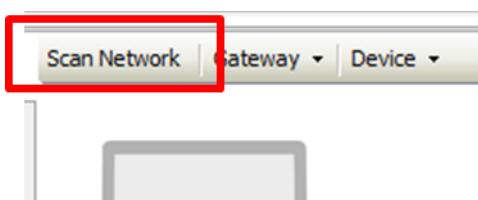


И выберите как минимум поле IP адреса. Введите IP адрес контроллера. При последующих нажатиях будут появляться следующие поля, которые можно не менять и не выводить.

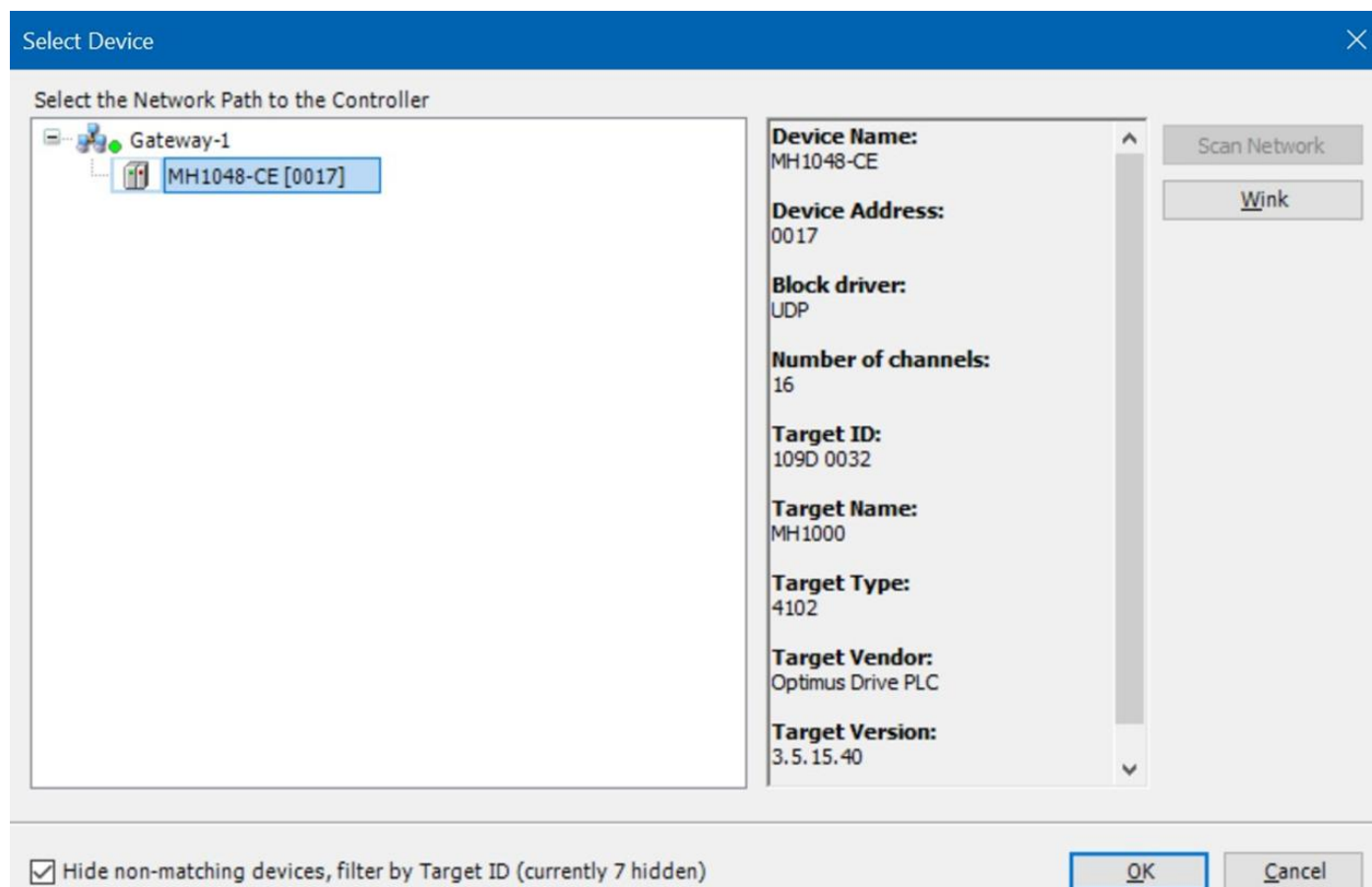
У контроллера IP адрес по умолчанию - **192.168.1.3**

Interface	Setting
 TCP interface	
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:

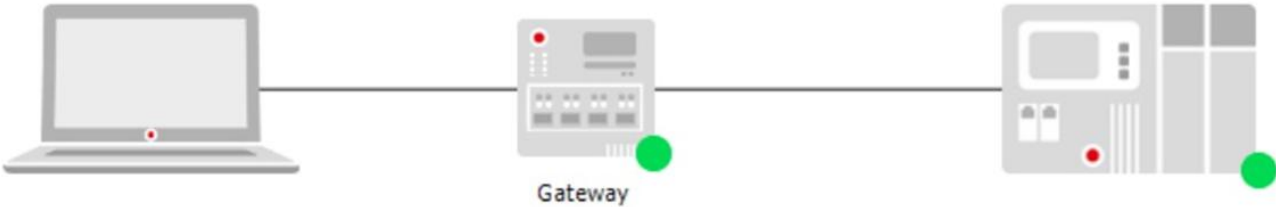


Нажмите OK и появится окно соединения с контроллером:

Scan Network

Gateway ▾

Device ▾



Gateway

Gateway-1 ▾

IP-Address:
localhost

Port:
1217

MH1048-CE (active) ▾

Device Name:
MH1048-CE

Device Address:
0017

Target ID:
109D 0032

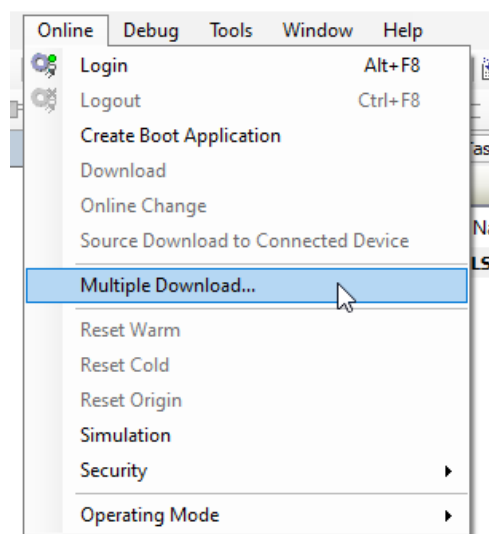
Target Type:
4102

Target Vendor:
Optimus Drive PLC

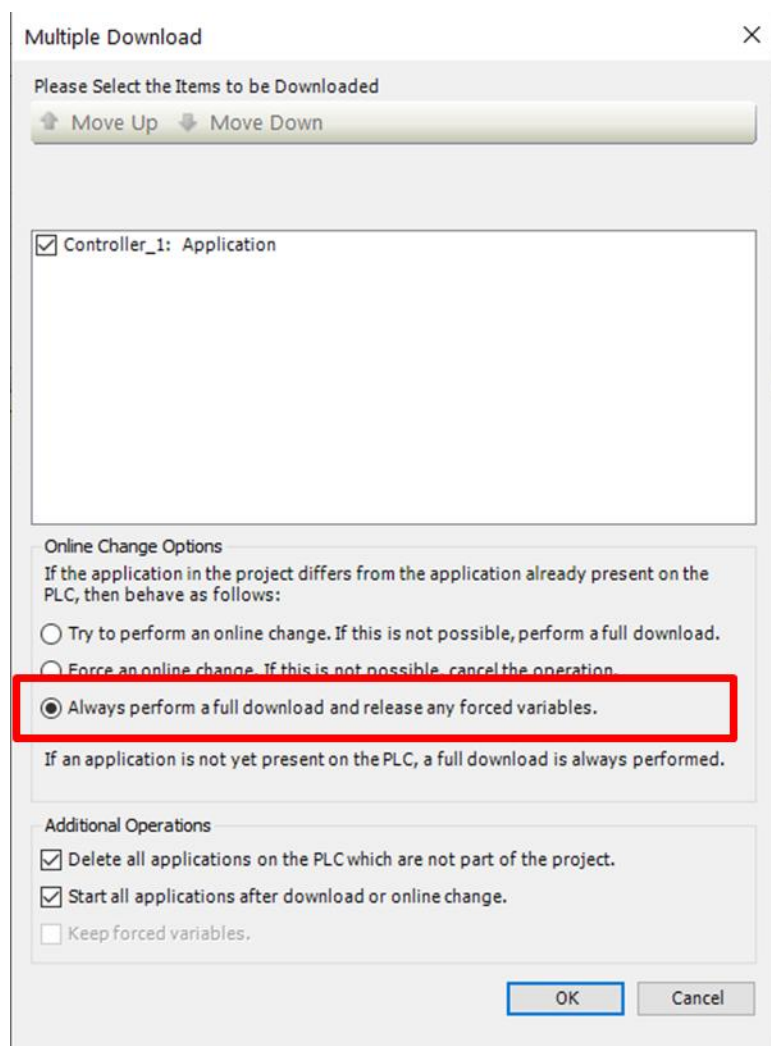
Target Version:
3.5.15.40

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:

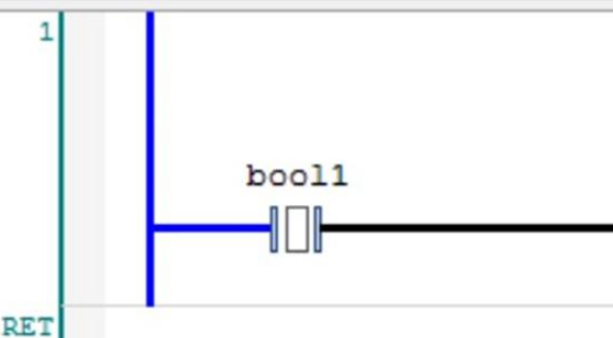
Devices

- Untitled1
 - Controller_1 [connected] (MH1048-CE)
 - PLC Logic
 - Application [run]**
 - Library Manager
 - General (PRG)
 - Task Configuration
 - MainTask
 - General
 - High_Speed_IO_Module (MHHighSpeed)
 - SoftMotion General Axis Pool

Controller_1 **General**

Controller_1.Application.General

Expression	Type	Value	Pre
bool1	BOOL	FALSE	
bool2	BOOL	FALSE	

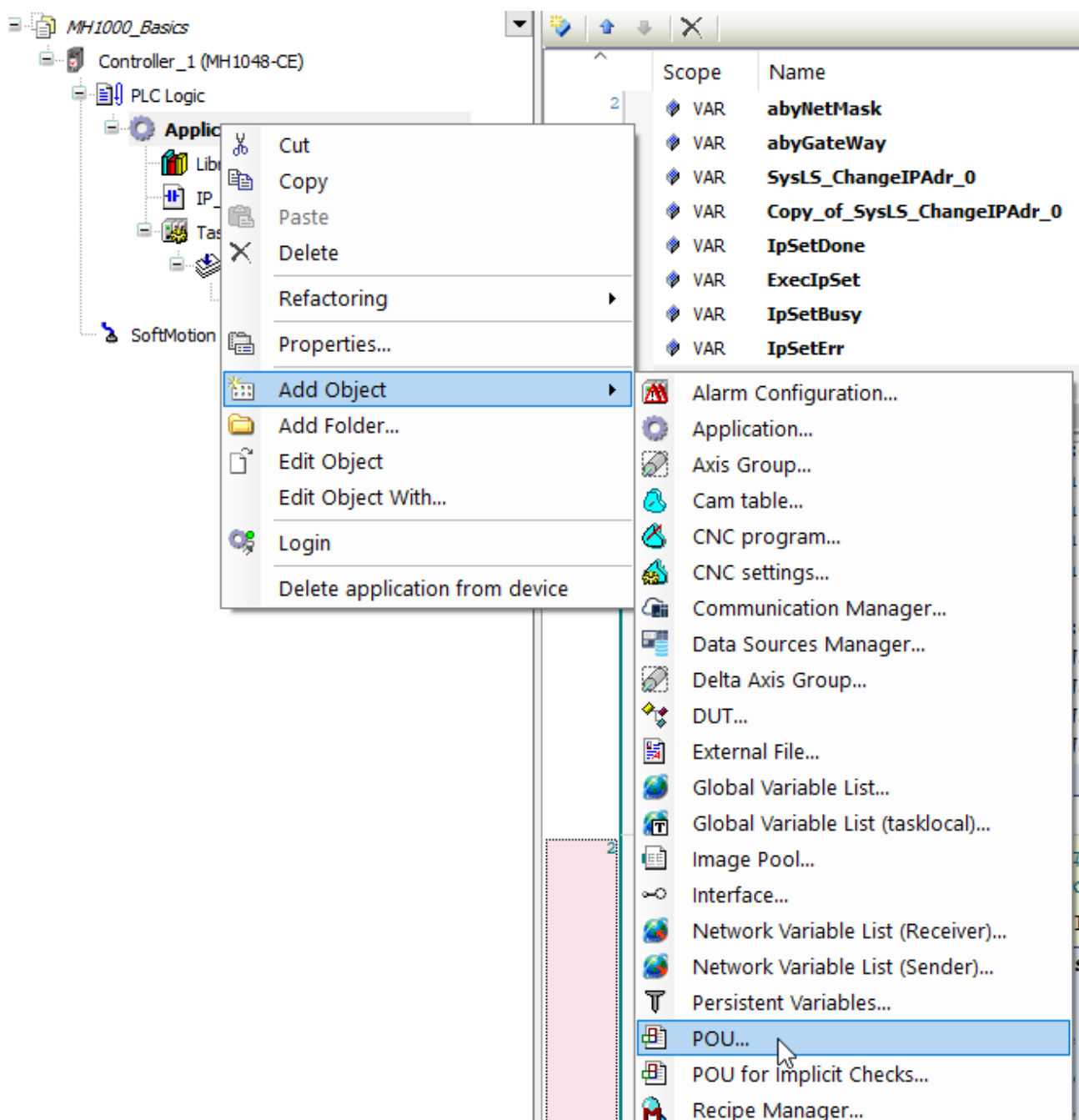


Для выхода из режима онлайн нажмите кнопку с красным крестиком:

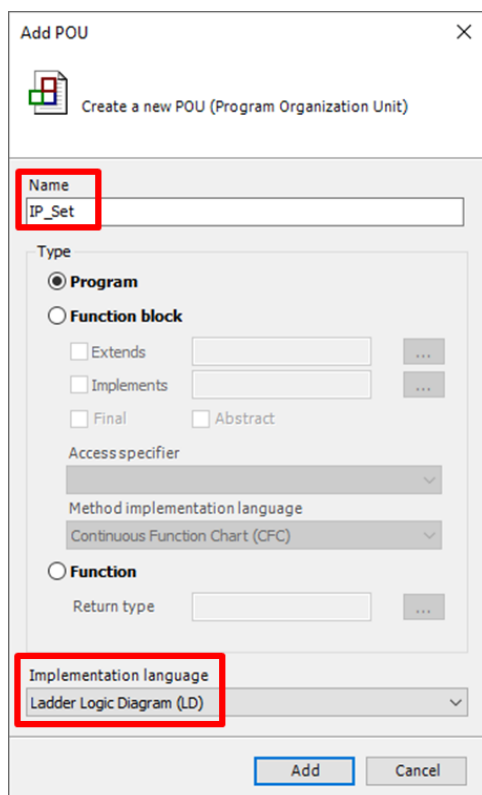


Изменение IP адреса контроллера из программы контроллера

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**:

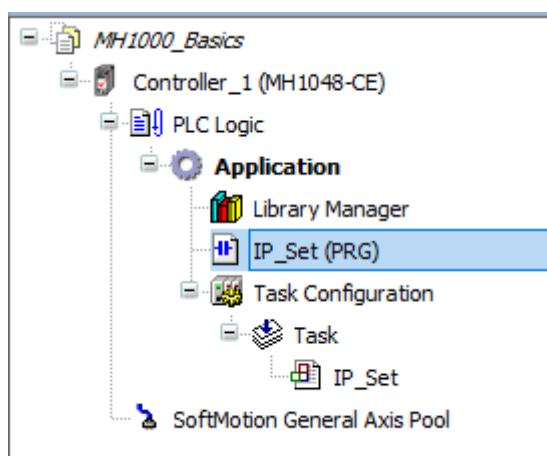


В открывшемся окне задайте название и язык программирования:

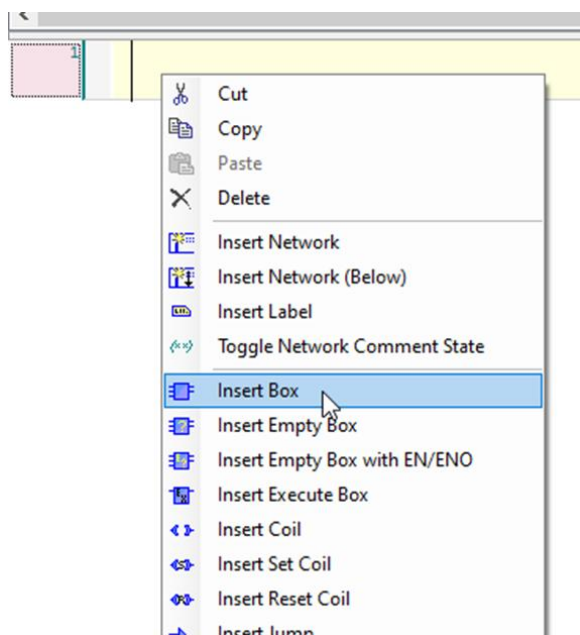


The 'Add POU' dialog box is shown. The 'Name' field is set to 'IP_Set'. The 'Type' is set to 'Program'. The 'Implementation language' is set to 'Ladder Logic Diagram (LD)'. The 'Add' button is highlighted.

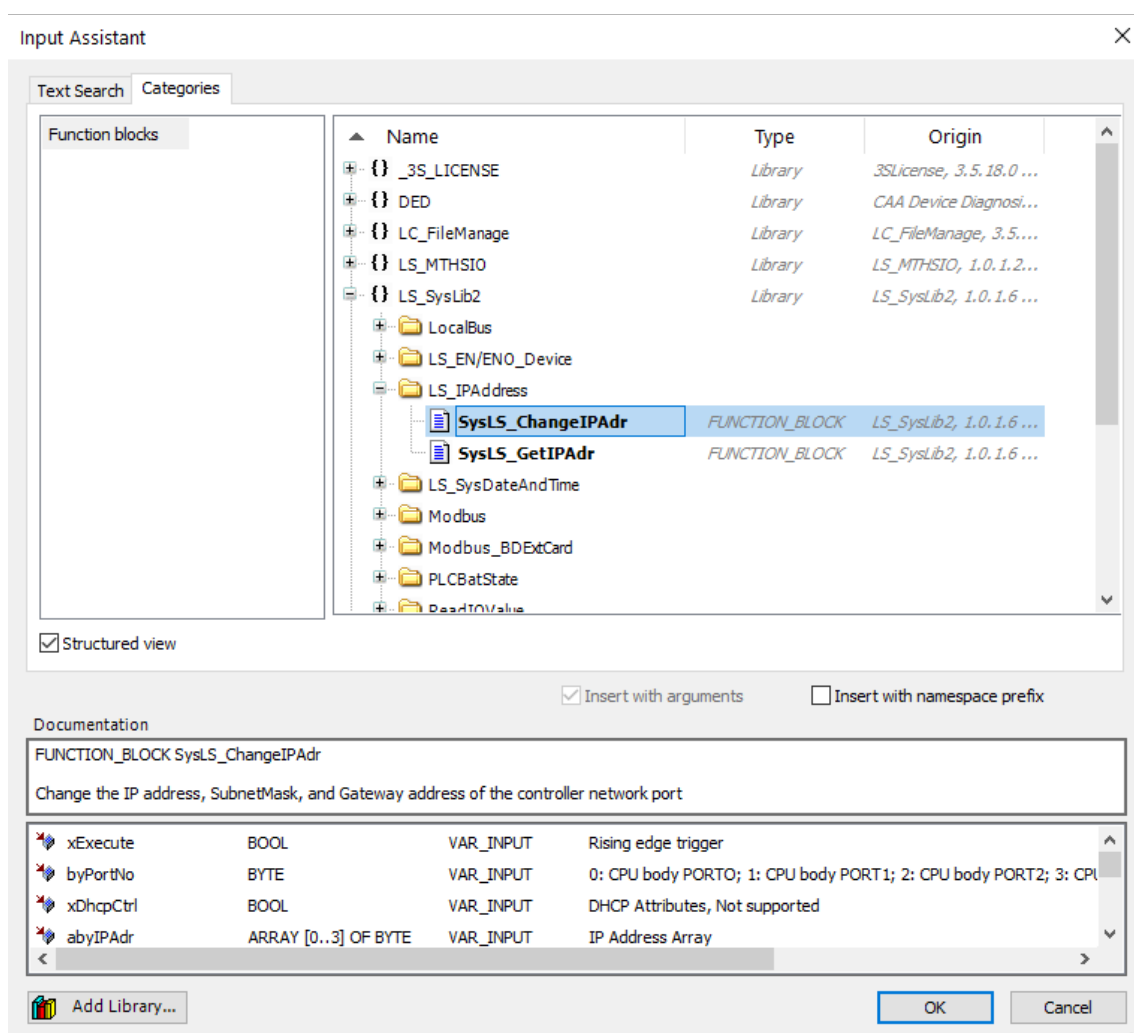
Нажмите кнопку **Add**. В древе проекта появится POU IP_Set



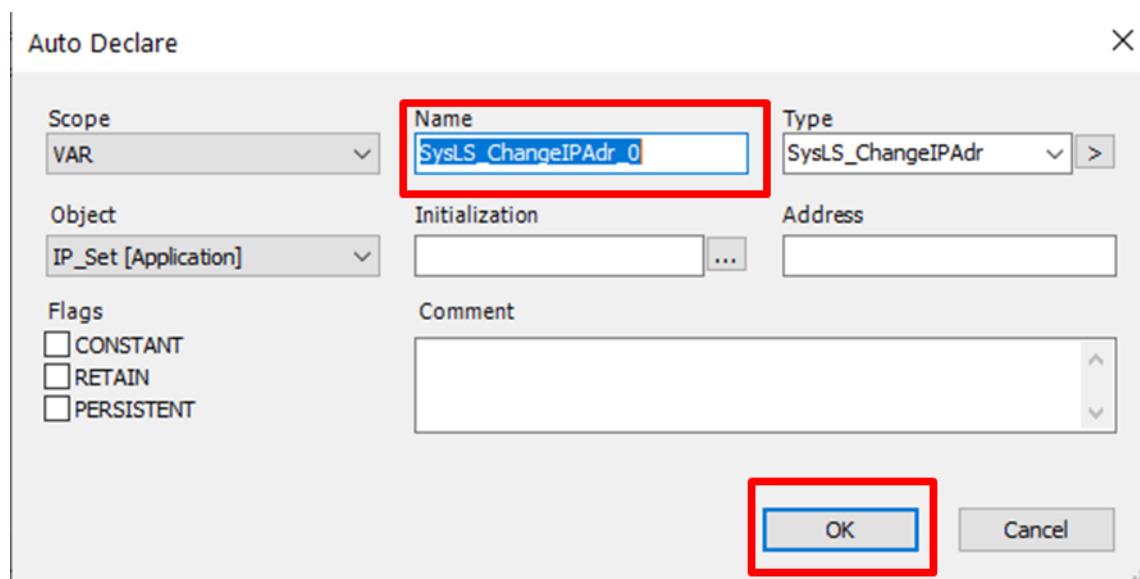
В окне созданного POU в поле ввода кода нажмите правой кнопкой мышки и выберите пункт **Insert Box**:



Откроется Мастер выбора команд. Нужно выбрать команду **SysLS_ChangeIPAdr** (библиотека LS_SysLib2)



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



The 'Auto Declare' dialog box is shown with the following fields:

- Scope:** VAR
- Name:** SysLS_ChangeIPAdr_0 (highlighted with a red box)
- Type:** SysLS_ChangeIPAdr
- Object:** IP_Set [Application]
- Initialization:** (empty field)
- Address:** (empty field)
- Comment:** (empty text area)
- Flags:**
 - ☐ CONSTANT
 - ☐ RETAIN
 - ☐ PERSISTENT
- Buttons:** OK (highlighted with a red box) and Cancel

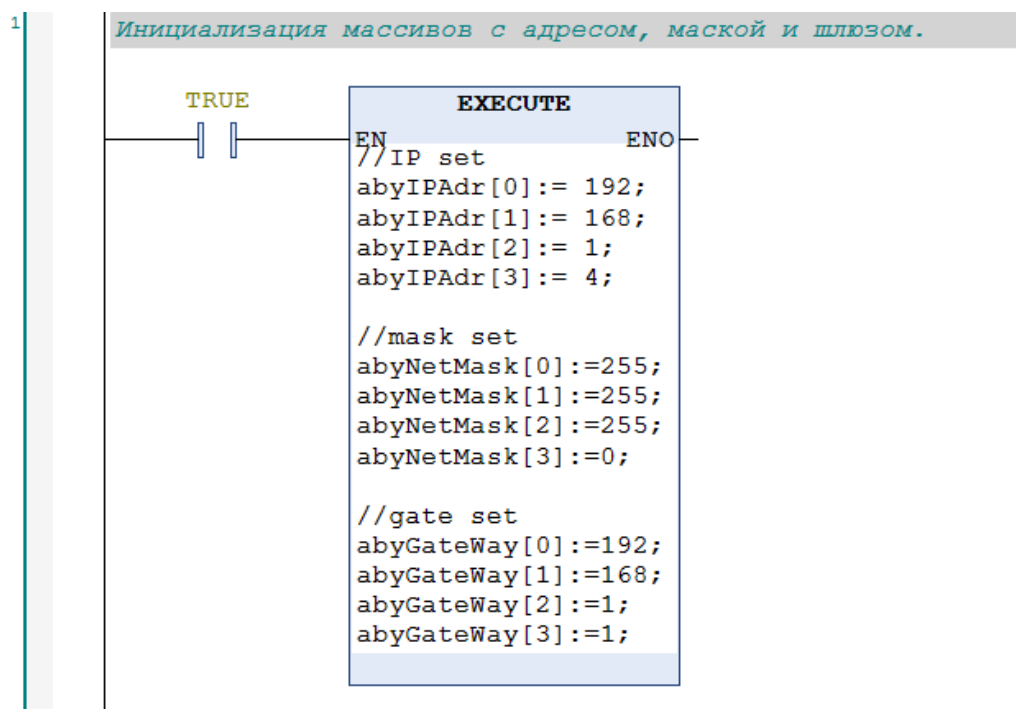
Появится переменная в таблице, а команда на строке кода:

	Scope	Name	Address	Data type	Initialization
1	VAR	SysLS_ChangeIPAdr_0		SysLS_ChangeIPAdr	

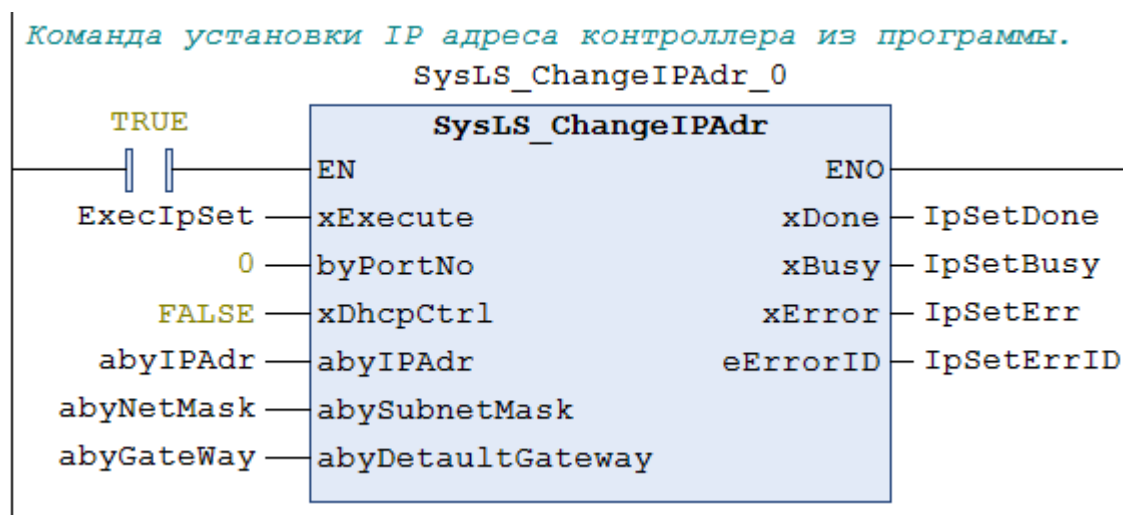
Адрес, маска и шлюз записываются в виде байтовых массив, которые необходимо объявить и инициализировать сразу в поле объявления переменных или в программе.

3	VAR	abyIPAdr	ARRAY[0..3] OF BYTE
4	VAR	abyNetMask	ARRAY[0..3] OF BYTE
5	VAR	abyGateWay	ARRAY[0..3] OF BYTE

Инициализация в программе:



Команда установки IP адреса, маски и шлюза:

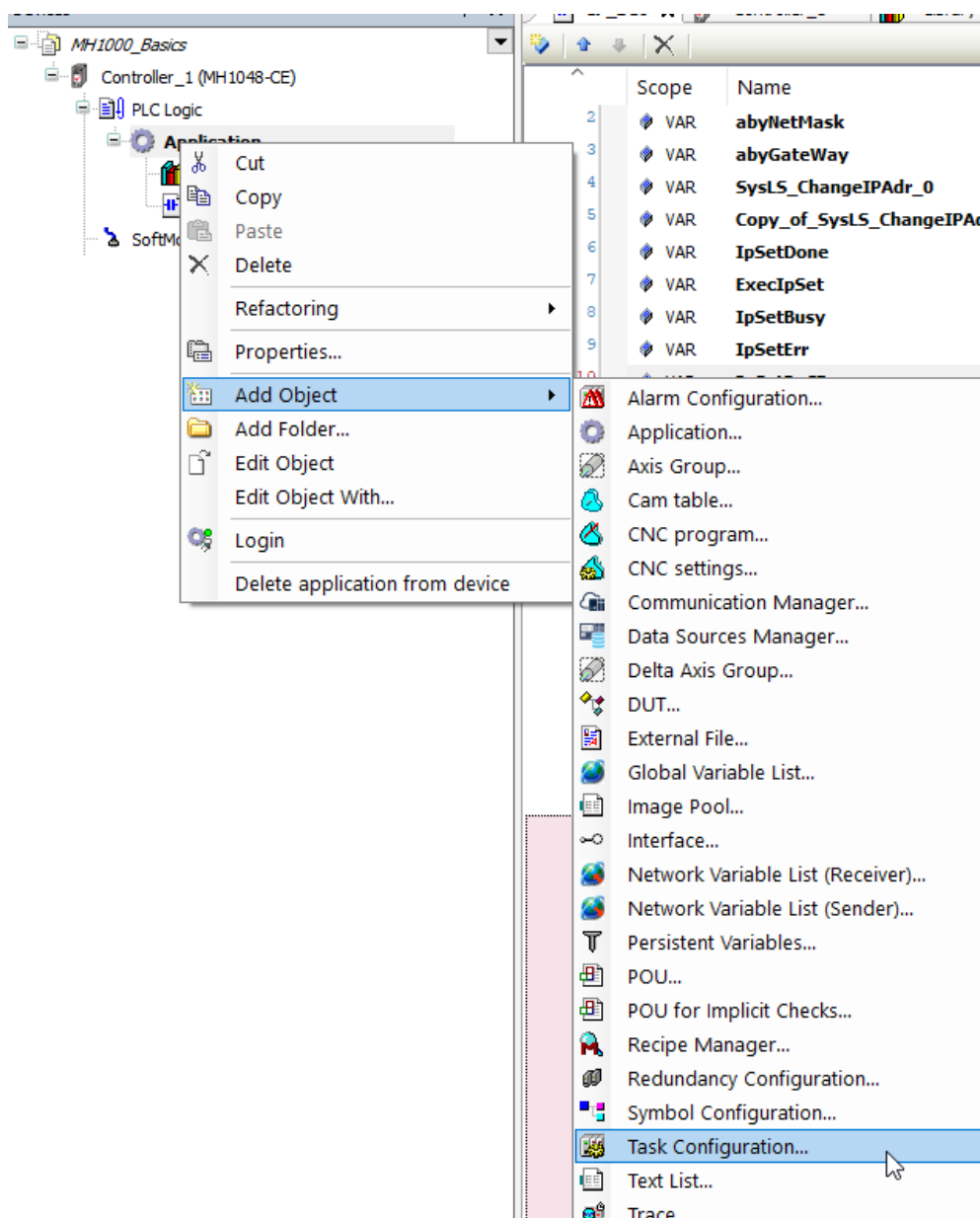


Номер порта задаётся на ножке **byPortNo**

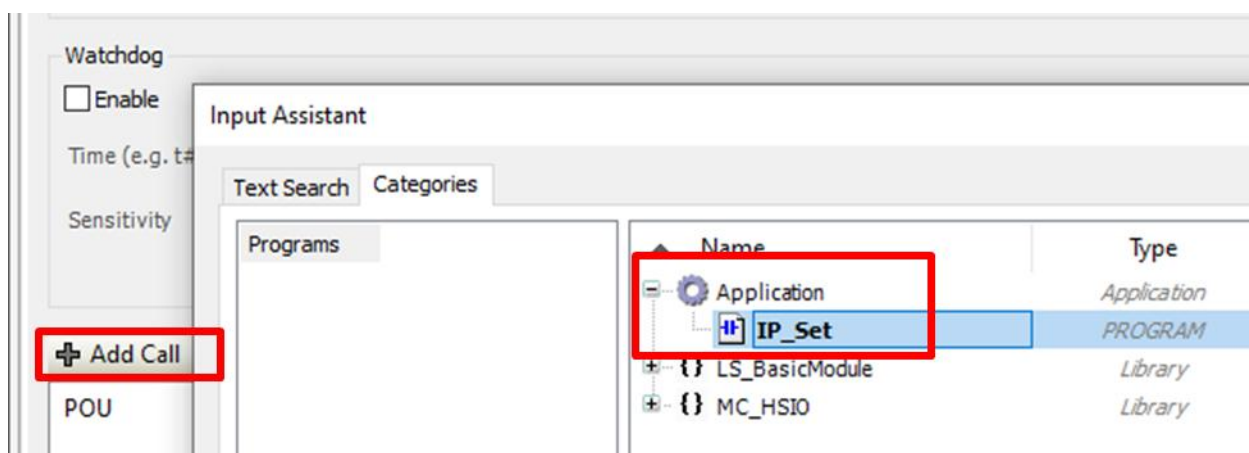
0 – LAN1

1 – LAN2

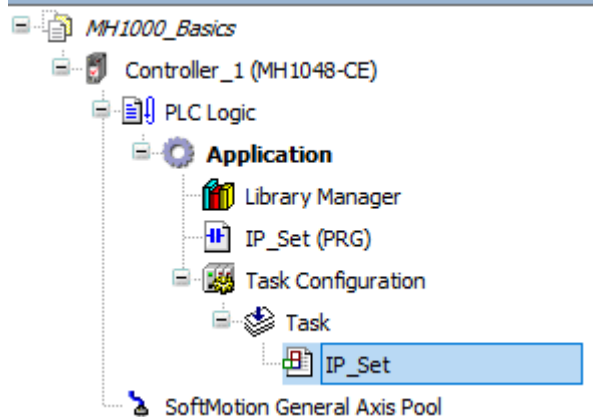
Далее необходимо добавить в проект Менеджер задач (Task Configuration):



и добавить в него POU IP_Set:



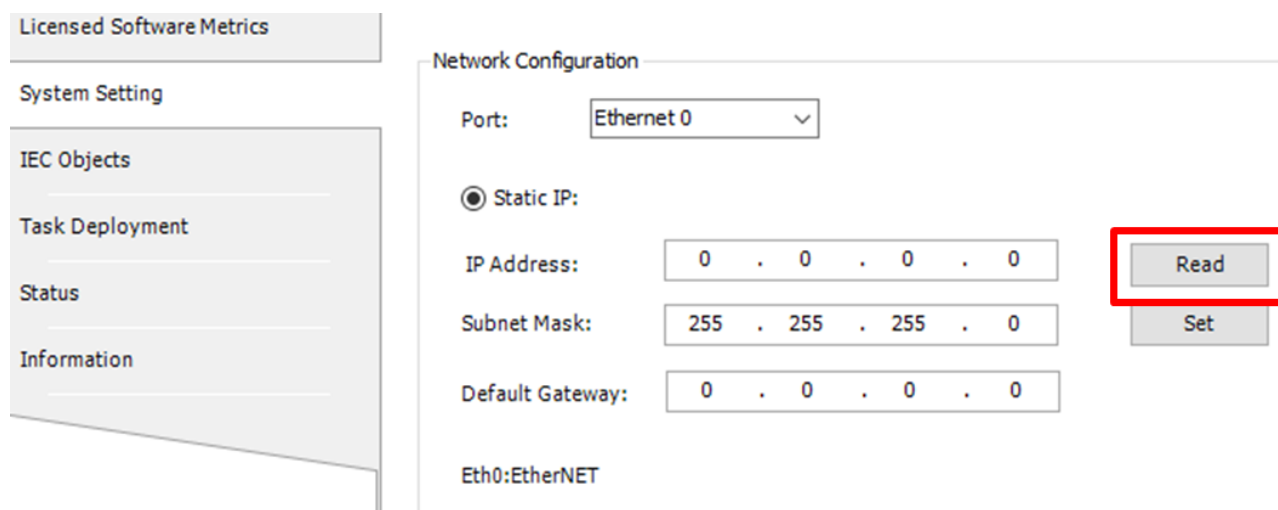
В древе проекта появится соответствующий пункт:



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

Изменение IP адреса контроллера из среды программирования

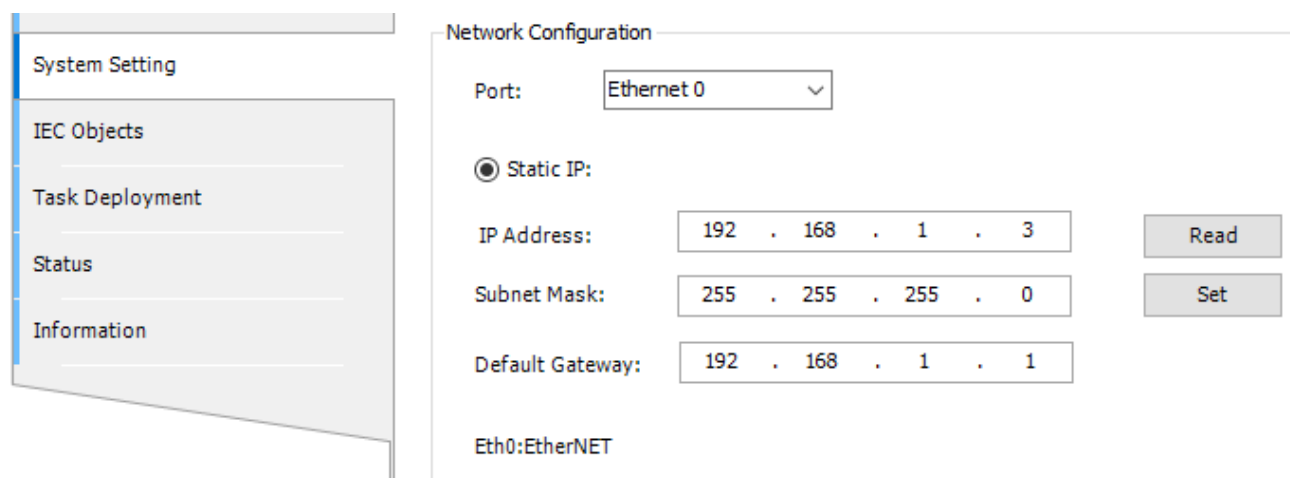
Для изменения IP адреса контроллера из среды программирования необходимо открыть вкладку Device и выбрать пункт System Setting, далее поле Network Configuration. Установите связь с контроллером (см. соответствующий раздел настоящего Руководства), и нажмите кнопку **Read**:



The screenshot shows the 'Network Configuration' window. On the left is a sidebar with menu items: 'Licensed Software Metrics', 'System Setting', 'IEC Objects', 'Task Deployment', 'Status', and 'Information'. The 'System Setting' menu is active. The main area is titled 'Network Configuration' and contains the following fields:

- Port:** A dropdown menu showing 'Ethernet 0'.
- Static IP:** A radio button that is selected.
- IP Address:** A text field with the value '0 . 0 . 0 . 0'.
- Subnet Mask:** A text field with the value '255 . 255 . 255 . 0'.
- Default Gateway:** A text field with the value '0 . 0 . 0 . 0'.
- Eth0:** A label with the value 'EtherNET'.
- Buttons:** 'Read' and 'Set' buttons are located to the right of the IP fields. The 'Read' button is highlighted with a red rectangle.

При наличии связи будет прочитан текущий IP адрес:

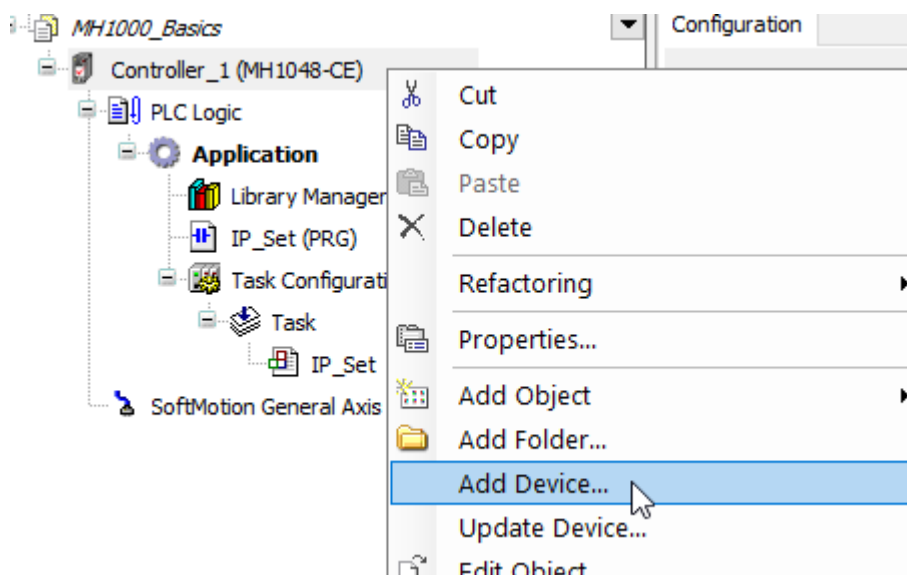


This screenshot shows the same 'Network Configuration' window after a successful connection. The 'Read' button has been clicked, and the IP address field now displays '192 . 168 . 1 . 3'. The 'Set' button is also visible below the 'Read' button. The sidebar and other fields remain the same as in the previous screenshot.

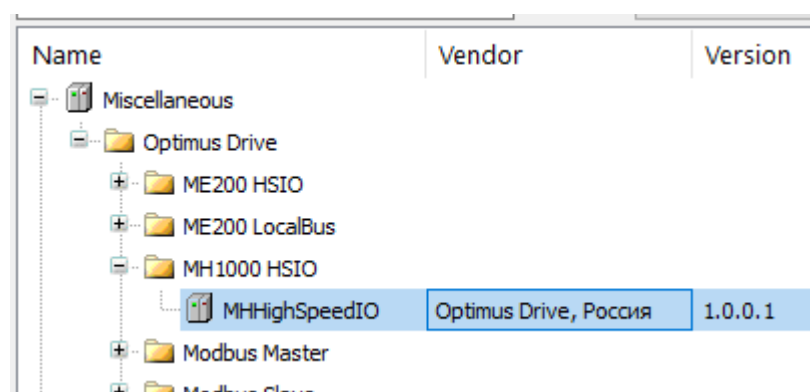
Далее задайте необходимый адрес и нажмите кнопку **Set**. После смены адреса произойдёт потеря связи между ПК и контроллером, поэтому необходимо заново установить связь ПК - контроллер.

Использование встроенных входов-выходов контроллера в обычном режиме

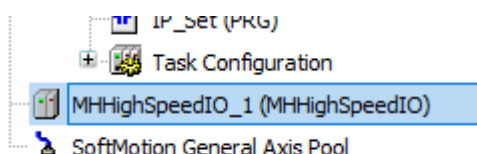
Для использования входов-выходов на ЦПУ в обычном режиме (не в импульсном) достаточно добавить в проект объект типа **High Speed IO** (идёт в составе пэкиджа) для чего щёлкните правой кнопкой мышки на контроллере в древе проекта и в отрывшемся меню выберите пункт **Add Device**:



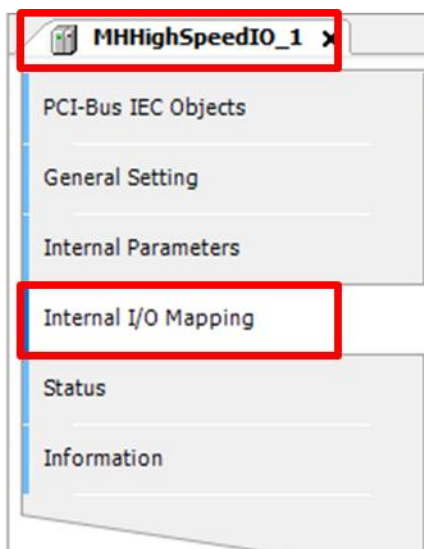
В появившемся окне выберите папку **Miscellaneous – Optimus Drive – MH1000 HSIO - MHHHighSpeedIO** и нажмите кнопку **Add Device**:



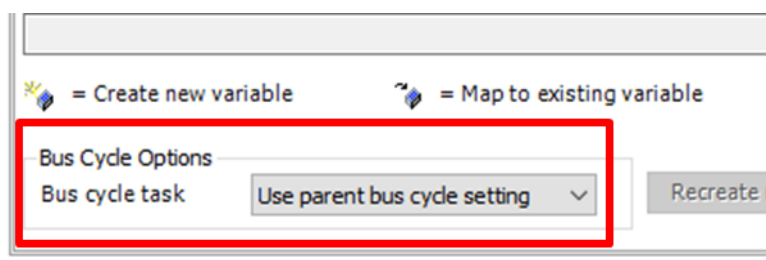
В древе проекта появится пункт **MHHHighSpeedIO**:



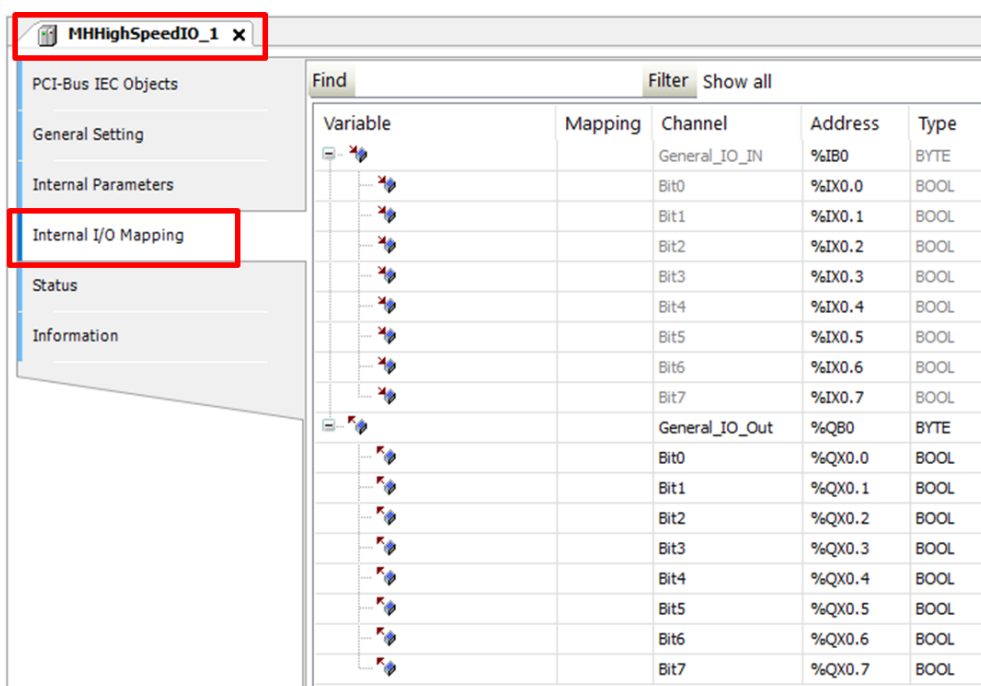
Адаптер **MHHighSpeedIO** привязывается к основной Задаче, указанной в настройках пункта Device. Как правило это EtherCAT Task.



В самом низу этой вкладки можно увидеть куда привязан адаптер (менять не рекомендуется!):



Для работы с входами и выходами на ЦПУ в обычном режиме достаточно открыть вкладку **Internal I/O Mapping**:



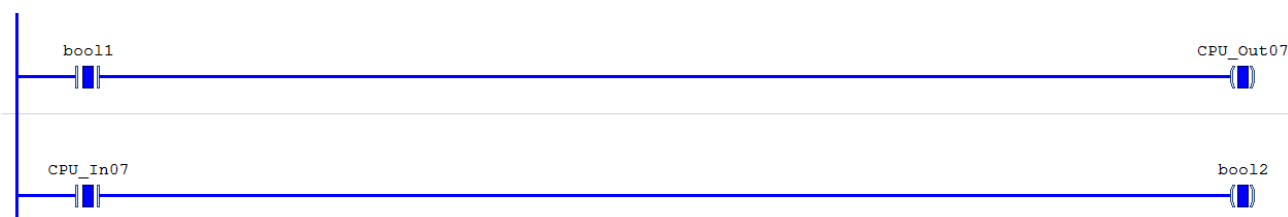
Развернуть нужный список и в колонке **Variable** задать имена переменных для входов и выходов:

		Bit6	%IX0.6	BOOL
	CPU_In07	Bit7	%IX0.7	BOOL

		Bit6	%QX0.6	BOOL
	CPU_Out07	Bit7	%QX0.7	BOOL

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.



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 ⁶⁴ -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 ⁶³ ~ (2 ⁶³ - 1)
	длинное целое число без знака	ULINT	64	0 ~ (2 ⁶⁴ -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 версией 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 MN1000 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 включить 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.

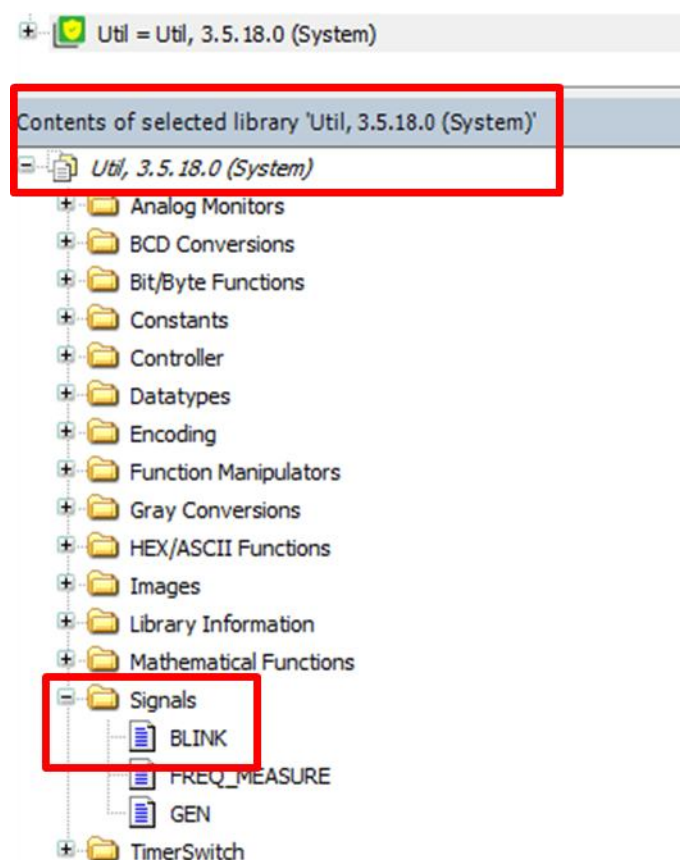
You have two options: Click through the topic tiles, or enter your search term directly into the search window.

	SEARCH IN CODESYS ONLINE HELP
----------------------	-------------------------------

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 МН1000 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
Сдвиг данных	SHR	FC	Побитовый сдвиг вправо
	SHL	FC	Побитовый сдвиг влево
	ROR	FC	Побитовое вращение вправо
	ROL	FC	Побитовое вращение влево

Категория инструкций	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 строинг
	WORD_TO_HEXSTRING	FC	Слово в шестнадцатеричный строинг
	DWORD_TO_HEXSTRING	FC	Двойное слово в шестнадцатеричный строинг

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 CAA File Library	FILE.Open	FB	Открыть файл
	FILE.Close	FB	Закрыть файл
	FILE.Write	FB	Записать файл
	FILE.Read	FB	Прочитать файл
	FILE.Delete	FB	Удалить файл
	FILE.Copy	FB	Копировать файл
	FILE.Rename	FB	Переименование файла
	FILE.SetPos	FB	Установить местоположение чтения и записи файла
	FILE.GetPos	FB	Получить местоположение чтения и записи файла
	FILE.GetSize	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	Наложенное 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
Управление интерполированными движением осей LS_IpoLib Library	LS_2AxisLine	FB	Двухосевая линейная интерполяция
	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

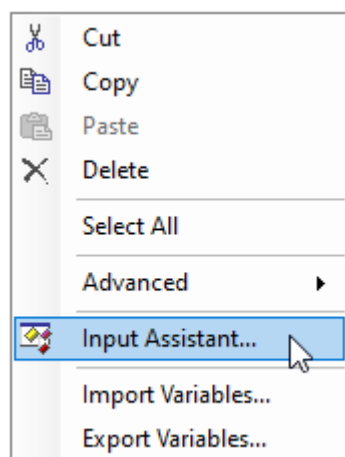
UDP сокет	UDP_Peer	FB	Открыть UDP сокет
	UDP_Receive	FB	Приём данных через UDP сокет
	UDP_Send	FB	Отправка данных через UDP сокет
Инструкции Ethernet/IP	IoDrvEtherNetIP_Diag	FB	Расширение для IoDrvEtherNetIP
	RemoteAdapter	FB	Расширения для LAT.Element
	Generic_Service	FB	Предоставляет generic service 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_MTHSIO	MC_Home_P	FB	Импульсная ось возвращается в исходное положение
	MC_MoveFeed	FB	
	LS_Counter	FB	Инструкция высокоскоростного счетчика
	LS_PresetValue	FB	Команда предустановленного значения счетчика
	LS_Compare	FB	Включение выхода по сравнению
	LS_EnableInterrupt	FB	Разрешение прерывания по входу
	LS_PresetValue	FB	Предустановленное значение счётчика
	LS_PWM	FB	Выдача ШИМ импульсов
	LS_TouchProbe	FB	Захват текущей позиции

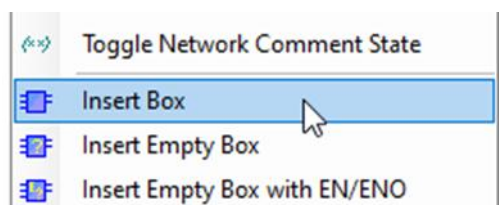
Категория инструкций	Название инструкции	FB/FC	Описание
	SysLS_GetIPAdr	FB	Прочитать IP адрес контроллера
	SysLS_ChangeIPAdr	FB	Установить IP адрес контроллера

Системные команды	GetBatState	FC	Прочитать уровень заряда батареи
	ColdResetApp	FC	«Холодная» перезагрузка контроллера
	WarmResetApp	FC	«Тёплая» перезагрузка контроллера

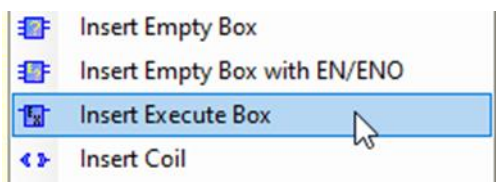
Для добавления в проект команды типа FB или FC на языке ST необходимо просто вызвать нажатием правой кнопки мышки и выбрать пункт **Input Assistant**:

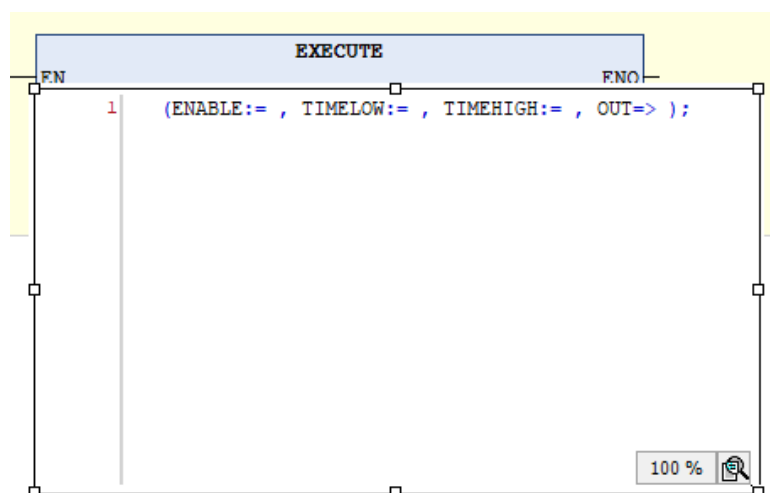


На языке LD команды типа FB вводятся через меню



а команды типа FC вводятся через

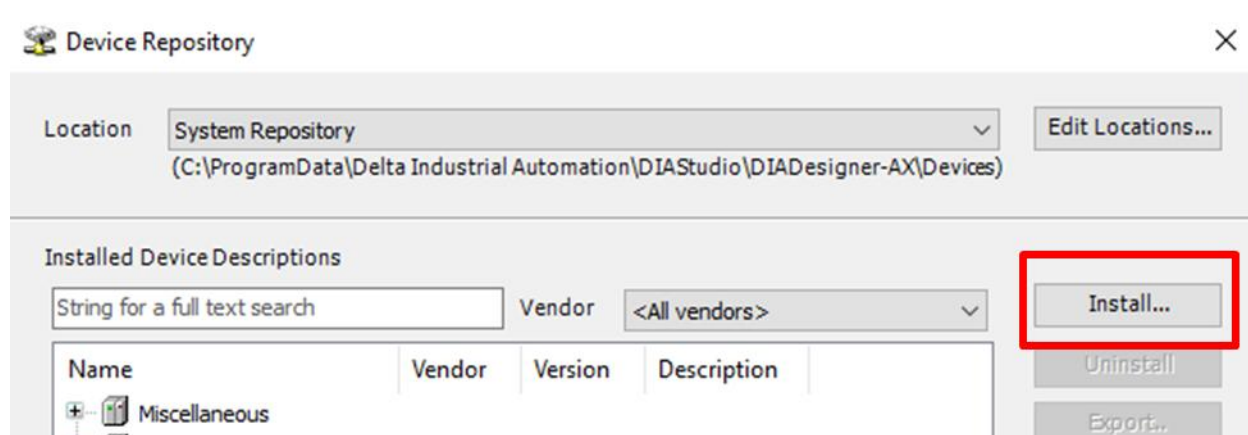
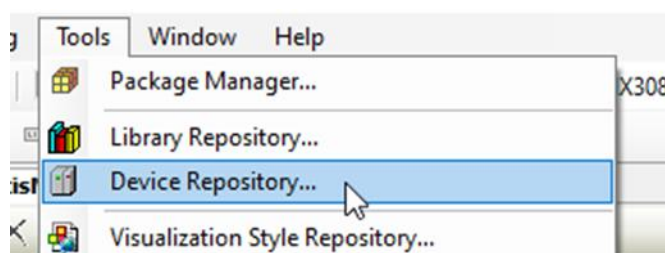




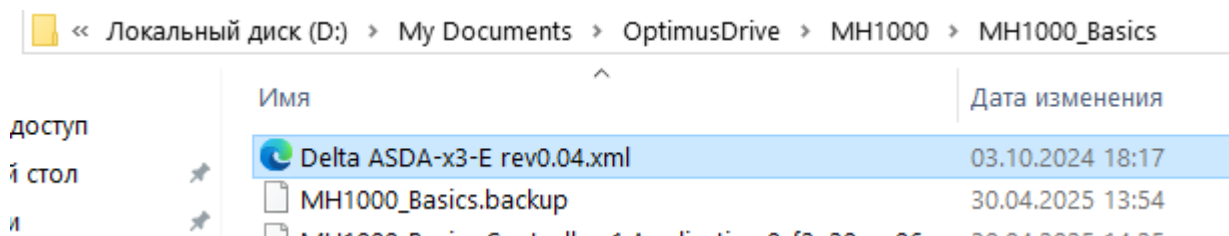
Добавление в проект сервопривода

Контроллеры серии MH1000 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.

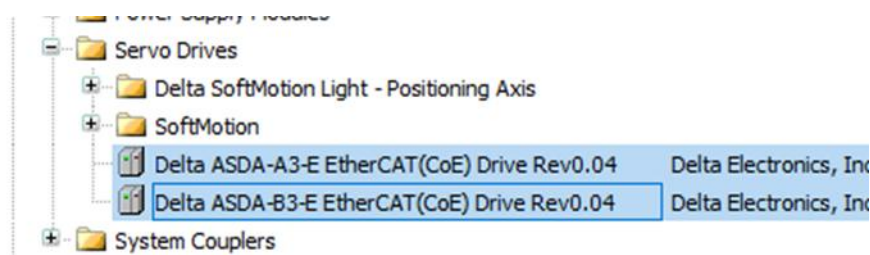
Для добавления в проект сервопривода сначала необходимо установить файл с описанием устройства в депозитарий (**Tools – Device Repository**):



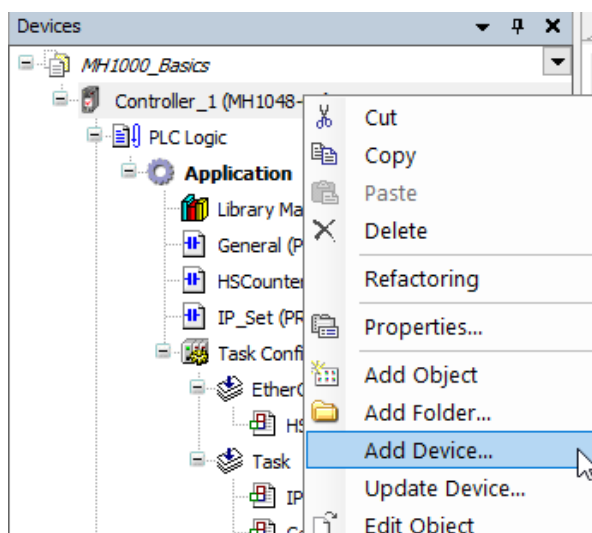
В качестве примера установим XML файл для сервопривода Delta ASD-B3-E:



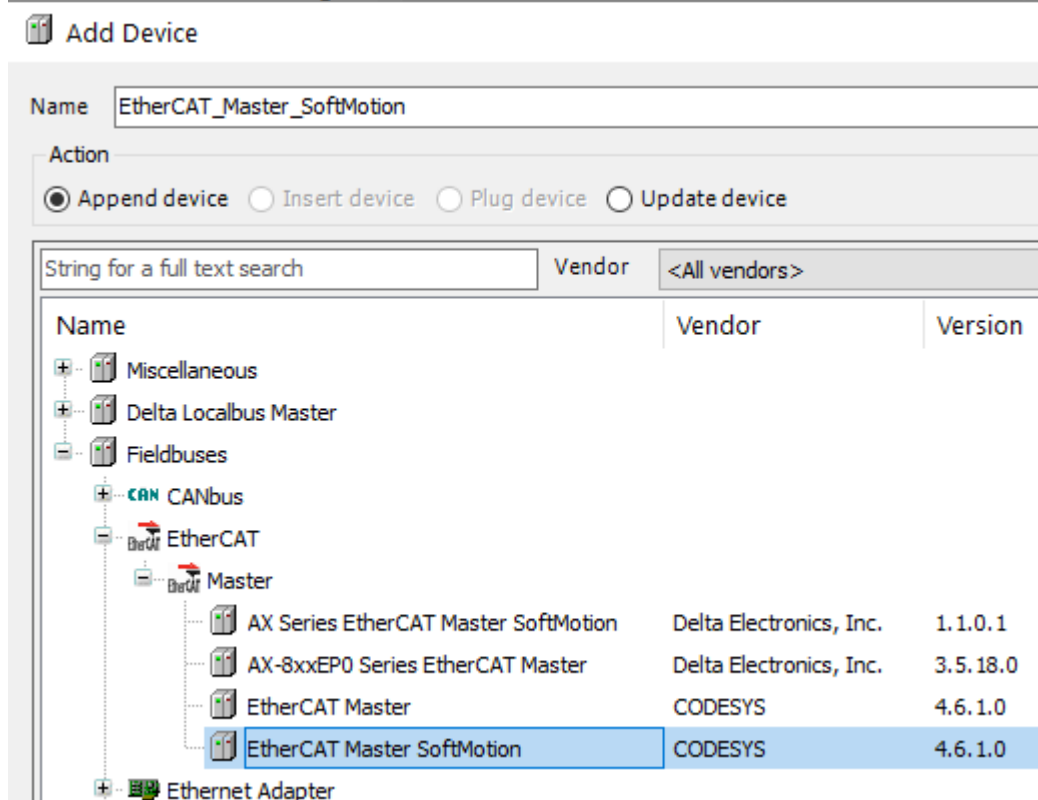
После установки файла в списке устройств появятся две записи:



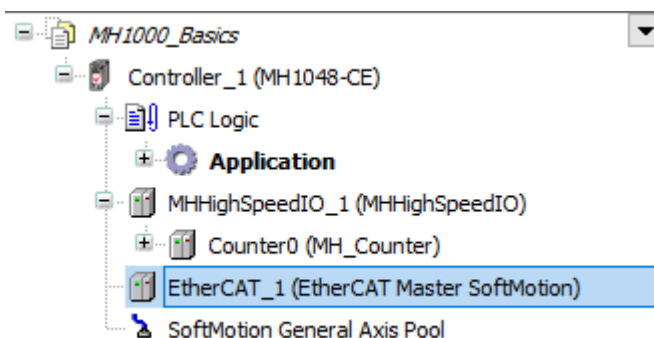
Далее необходимо добавить в проект EtherCAT адаптер. Для этого щёлкните правой кнопкой на названии контроллера и в меню выберите пункт **Add Device**:



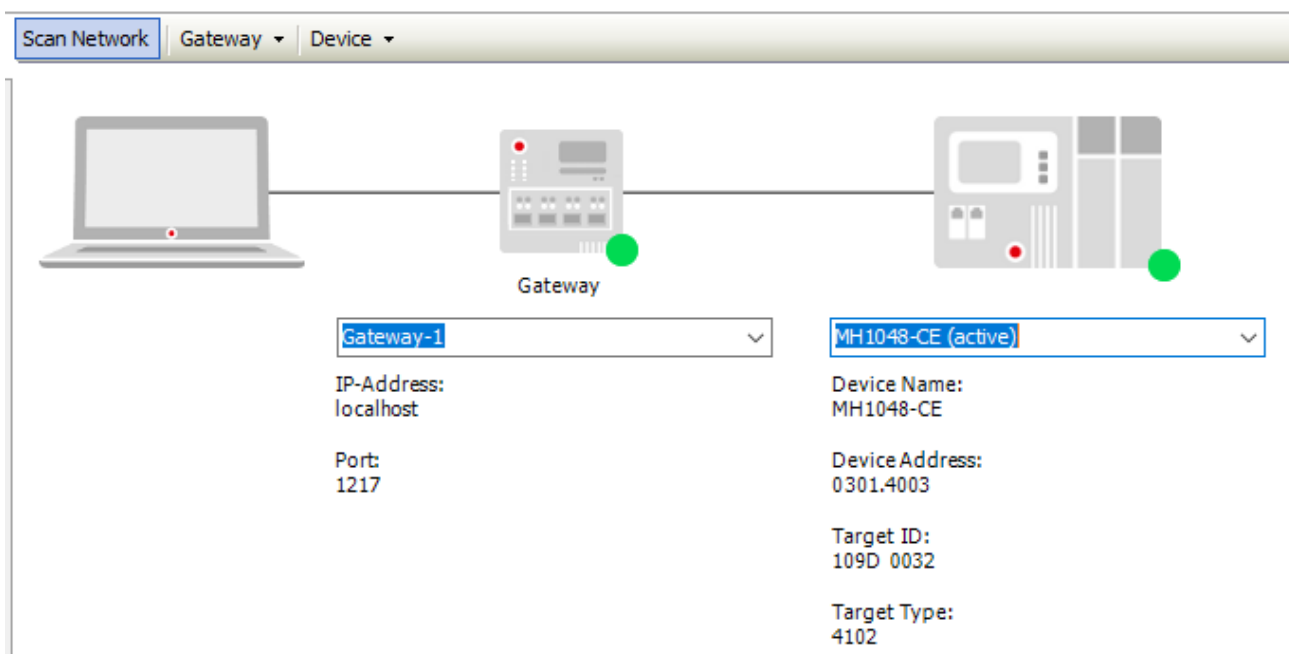
В открывшемся окне выберите пункт **EtherCAT - Master - EtherCAT Master SoftMotion**:



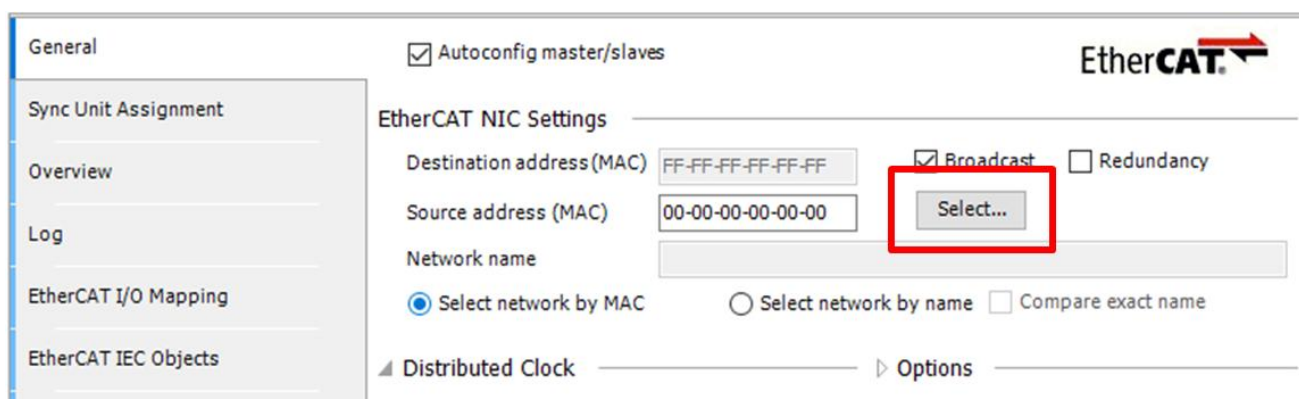
В древе проекта появится пункт:



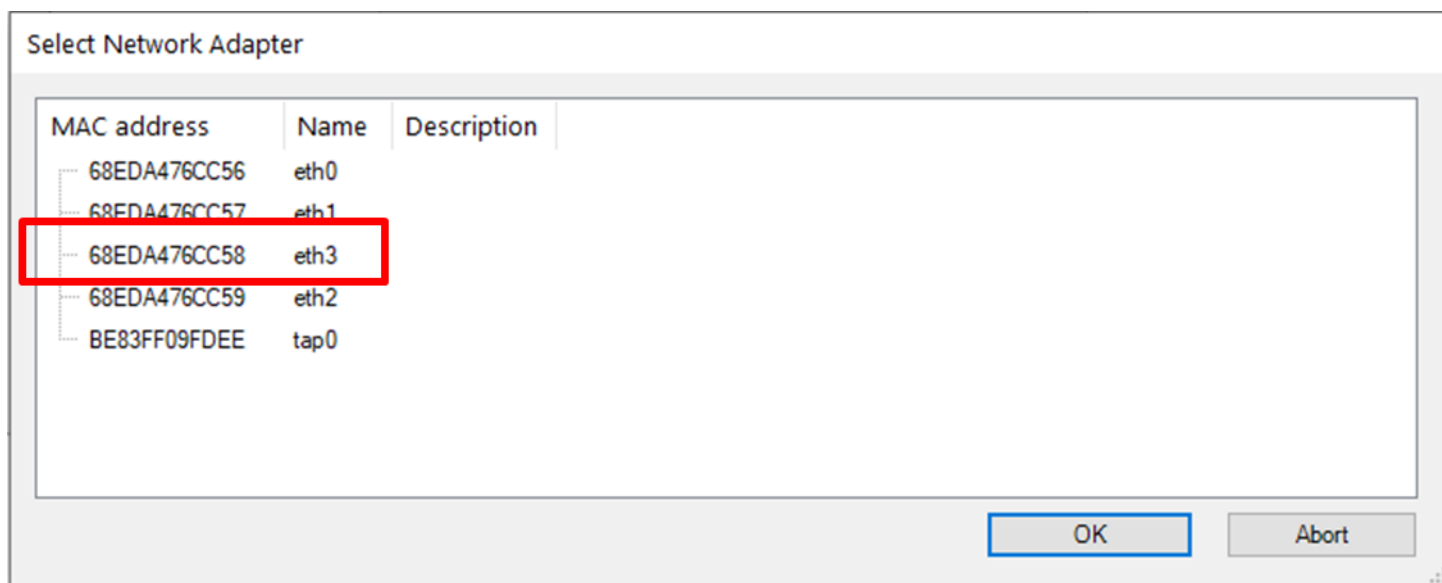
Далее необходимо назначить в проекте Мастера сети EtherCAT. В нашем примере это будет контроллер МН1000, поэтому необходимо установить с ним связь (см. соответствующий раздел данного Руководства):



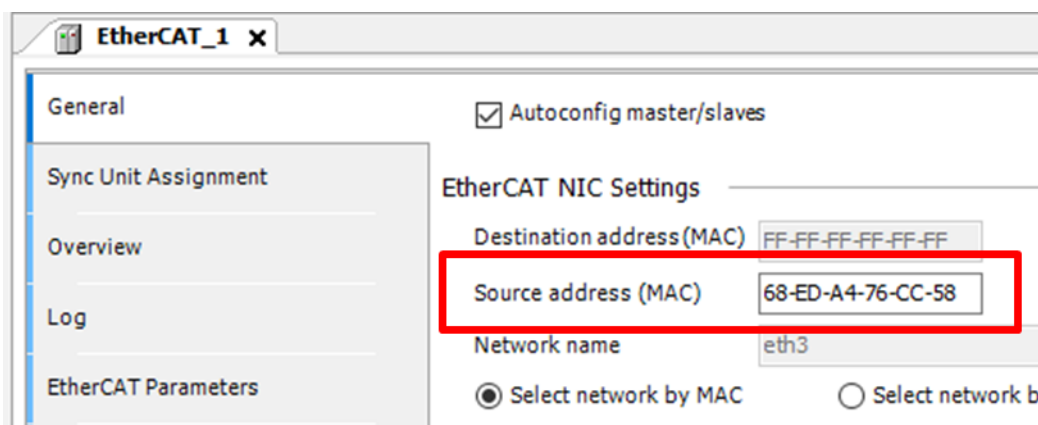
Далее щёлкните дважды левой кнопкой мышки на пункте **EtherCAT Master** и в открывшейся вкладке выберите пункт **General** и нажмите кнопку **Select**:



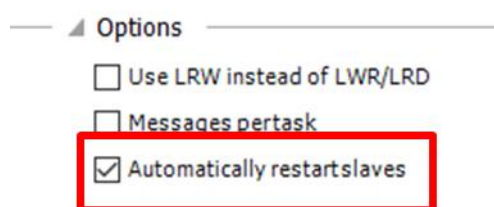
В открывшемся окне выберите пункт (MAC адрес у каждого контроллера свой):



MAC адрес должен установиться в данном поле:

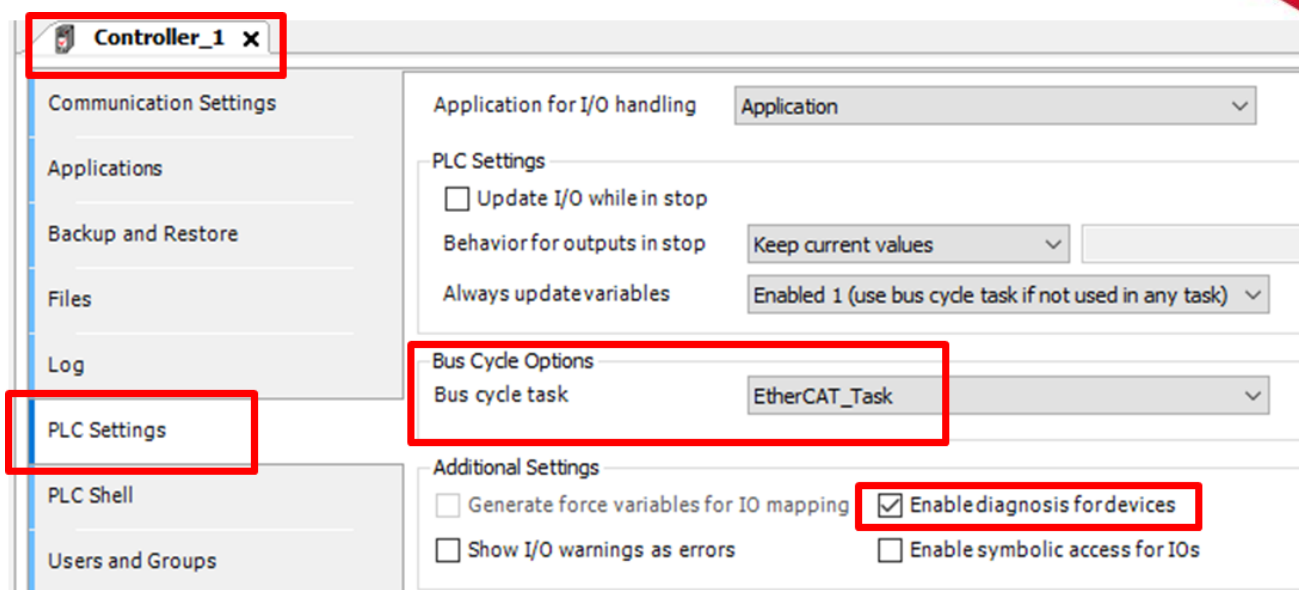


В этой же вкладке разверните пункт **Options** и поставьте флажок **Automatically restart slaves**:

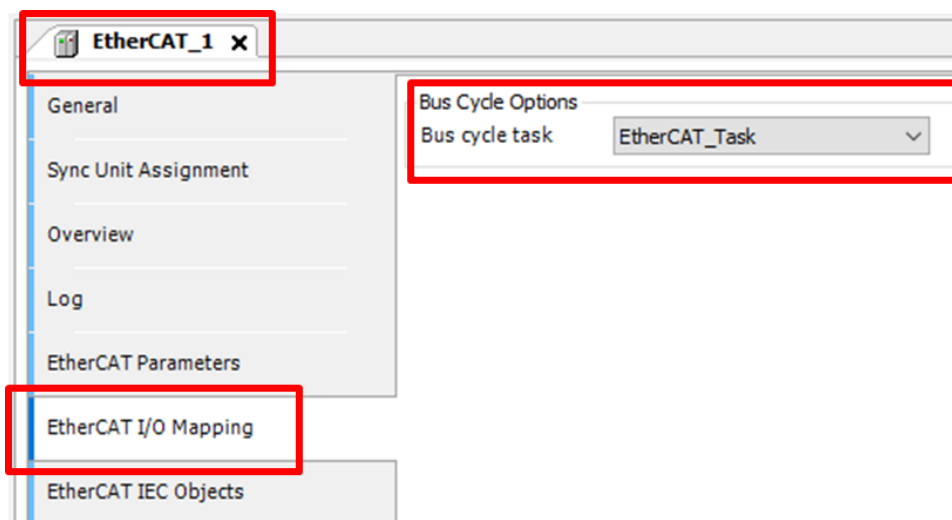


Далее необходимо активировать задачу EtherCAT_Task. Для этого сделайте следующие настройки:

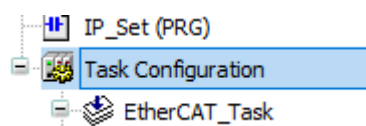
Во вкладке Контроллера:



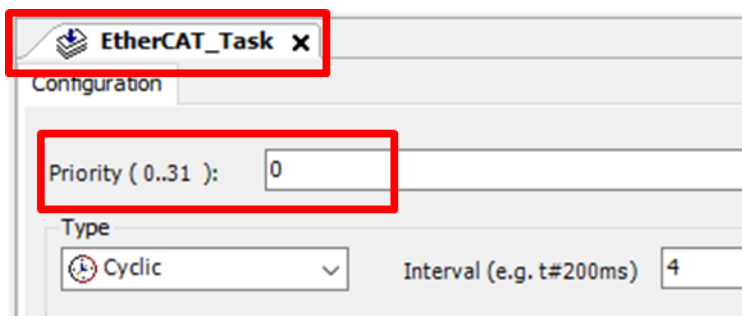
Во вкладке EtherCAT_Master:



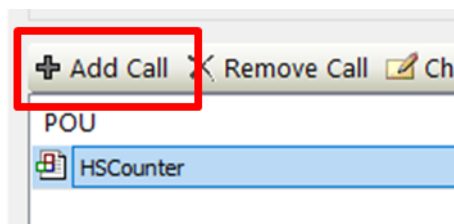
Далее в древе проекта во вкладке Менеджера Задач (Task Configuration):



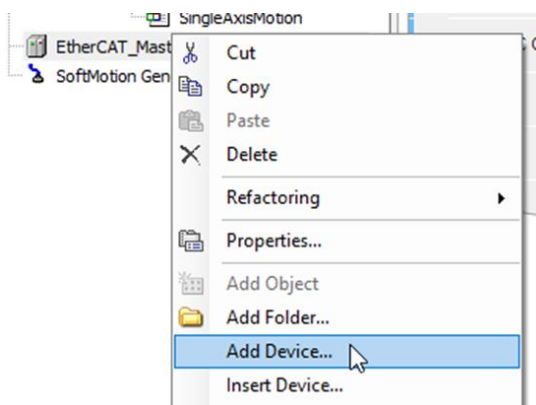
двойным щелчком левой кнопки мышки откройте задачу EtherCAT_Task и выставить ей высший приоритет (0):



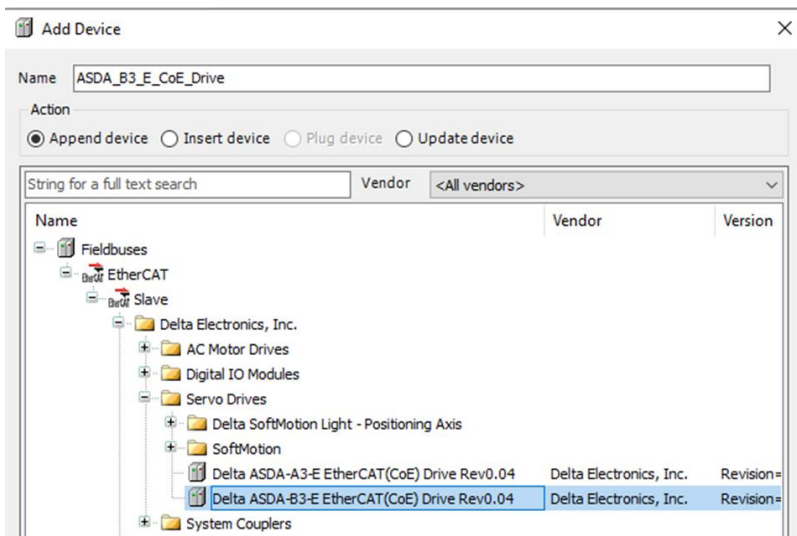
привязать нужные POU:



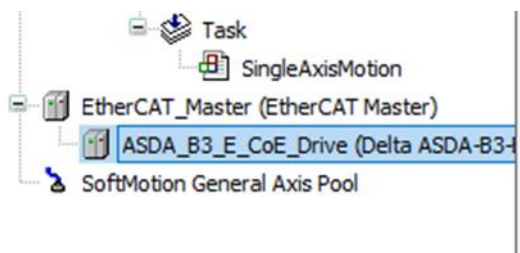
Далее необходимо добавить в проект сервопривод. Для этого щёлкните правой кнопкой мышки на пункте **EtherCAT Master** и в открывшемся окне выберите пункт **Add Device**:



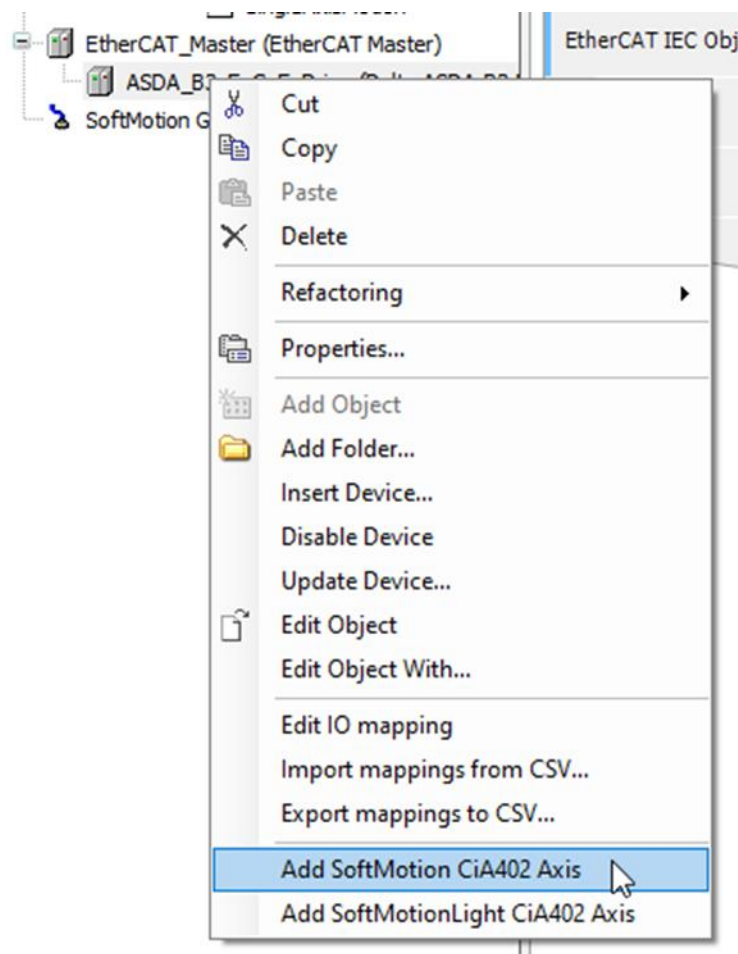
В открывшемся окне выберите сервопривод Delta ASD-B3-E:



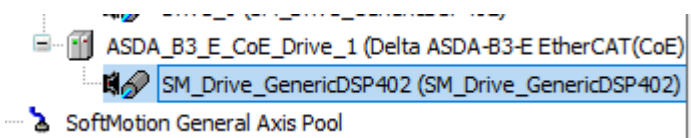
В древе проекта появится пункт:



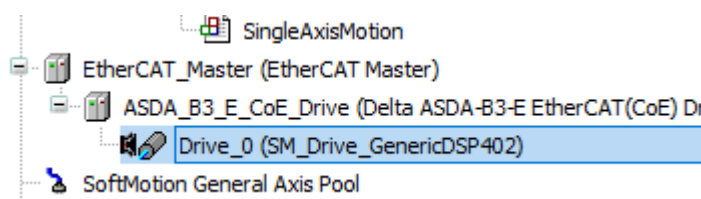
Далее к данному узлу необходимо добавить ось движения стандарта CiA402. Для этого щёлкните правой кнопкой мышки на пункте **ASDA_B3_E_CoE_Drive** и выберите пункт **Add SoftMotion CiA402 Axis**:



Появится пункт:

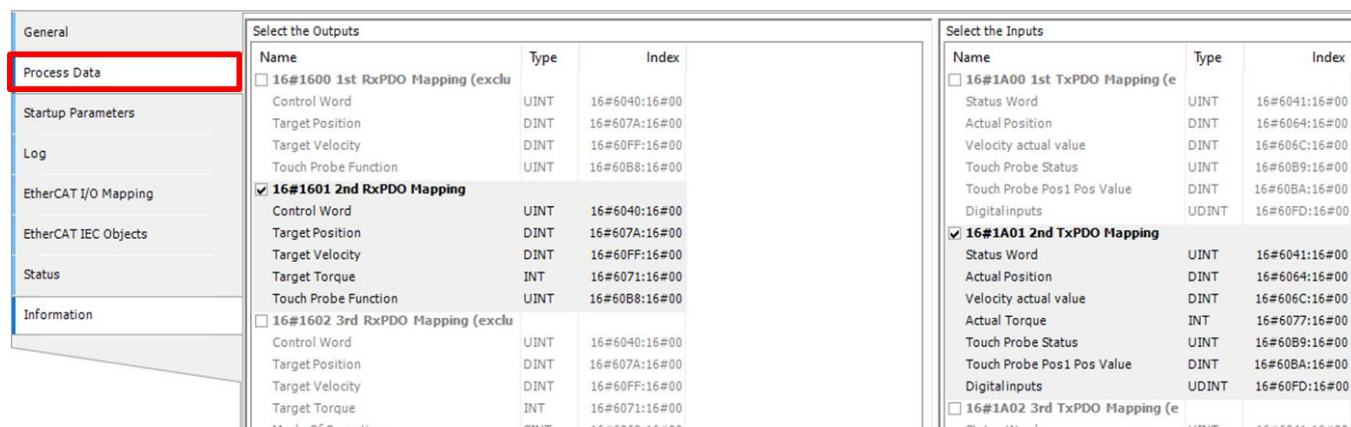


Название оси можно переименовать, например в Drive_0:



Далее необходимо произвести настройку узла и оси движения.

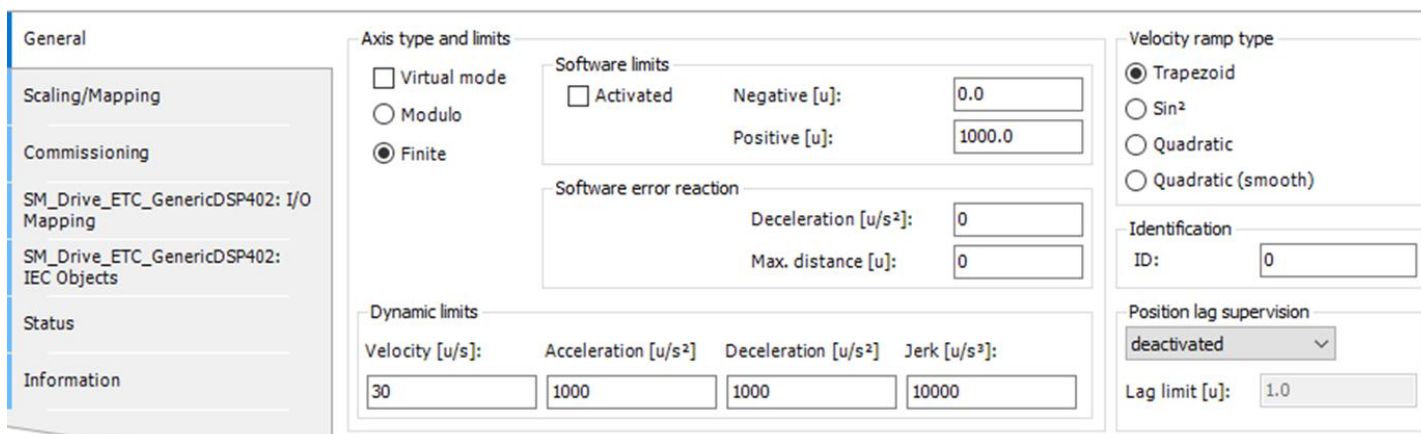
Для настройки узла щёлкните на пункте **ASDA_B3_E_CoE_Drive** два раза левой кнопкой мышки и в отрывшейся вкладке выберите пункт **Process Data**, в котором отметьте нужные наборы PDO:



Name	Type	Index
☐ 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
☒ 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
☐ 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
☐ 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
☒ 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
☐ 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]: 0.0
 Positive [u]: 1000.0

Software error reaction

Deceleration [u/s²]: 0
 Max. distance [u]: 0

Dynamic limits

Velocity [u/s]: 30
 Acceleration [u/s²]: 1000
 Deceleration [u/s²]: 1000
 Jerk [u/s³]: 10000

Velocity ramp type

☒ Trapezoid
☐ Sin²
☐ Quadratic
☐ Quadratic (smooth)

Identification

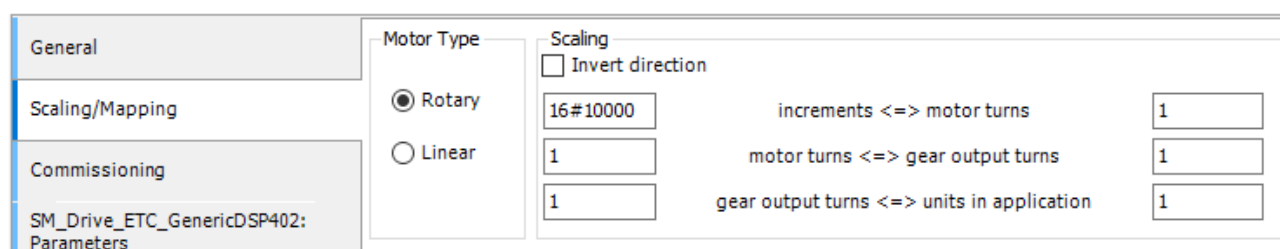
ID: 0

Position lag supervision

deactivated
 Lag limit [u]: 1.0

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**:



Motor Type

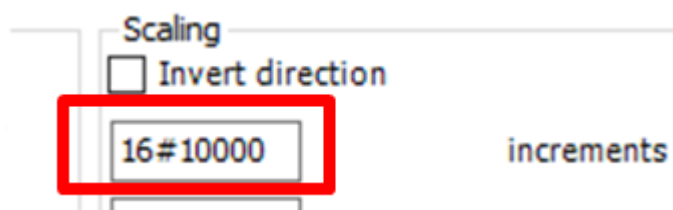
☒ Rotary
☐ Linear

Scaling

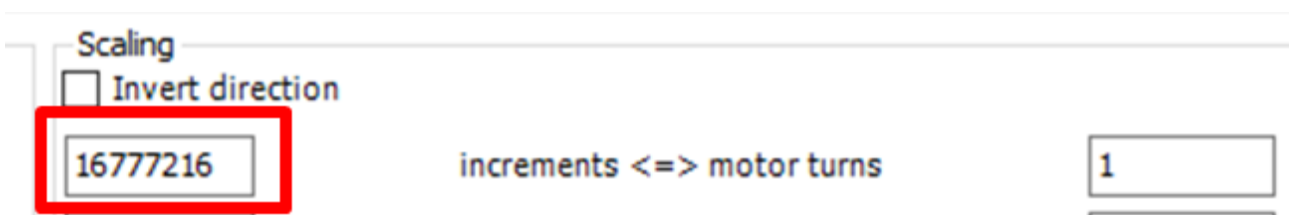
☐ Invert direction

16#10000 increments <=> motor turns 1
 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.



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).

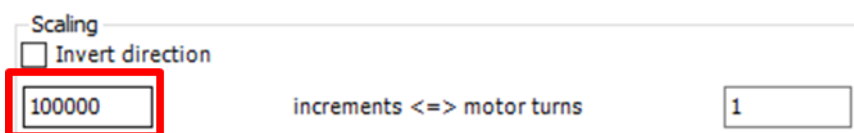


Если к примеру установить в приводе вот такие числитель и знаменатель:

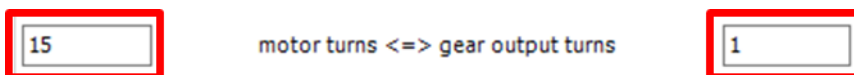
Числитель P1-044 = 16 777 216

Знаменатель P1-045 = 100 000

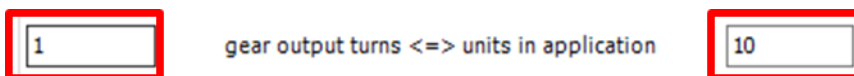
то мы получим число 100 000 импульсов на оборот, которое нужно будет ввести в поле:



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



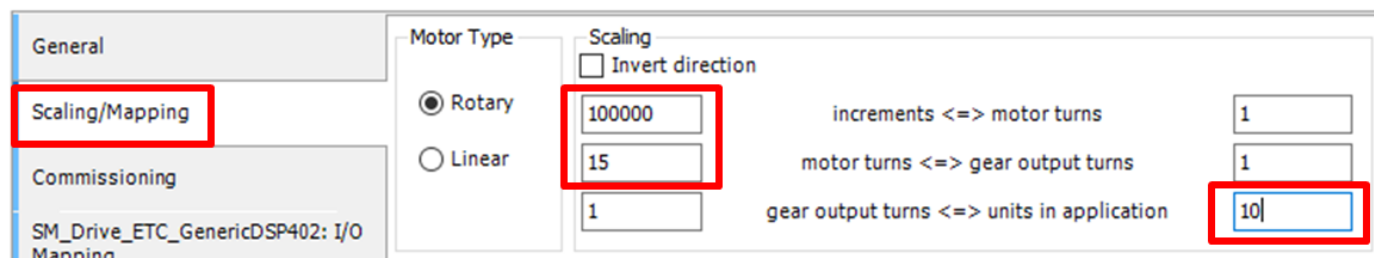
Во многих случаях редуктор не вводят, а учитывает его в конечном перемещении, которое вводится в третьей строчке. В нашем примере это 10 единиц.



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:

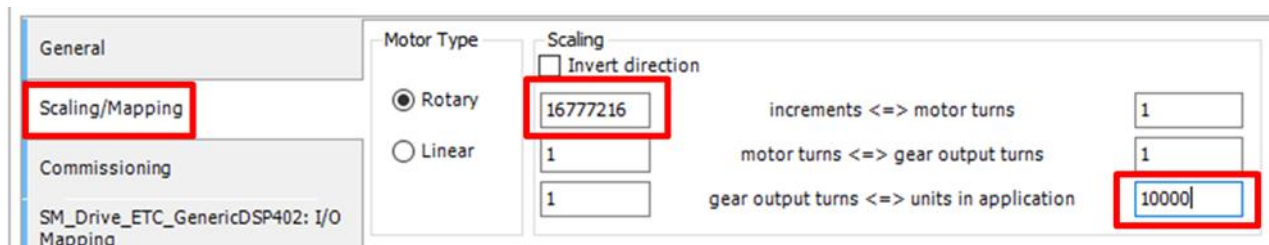


Для улучшения точности, перемещение конечного механизма на 1 оборот лучше измерять прямым способом высокоточным измерительным инструментом (микрометром) и уже на основе объективных данных устанавливать линейные единицы пользователя.

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:



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.

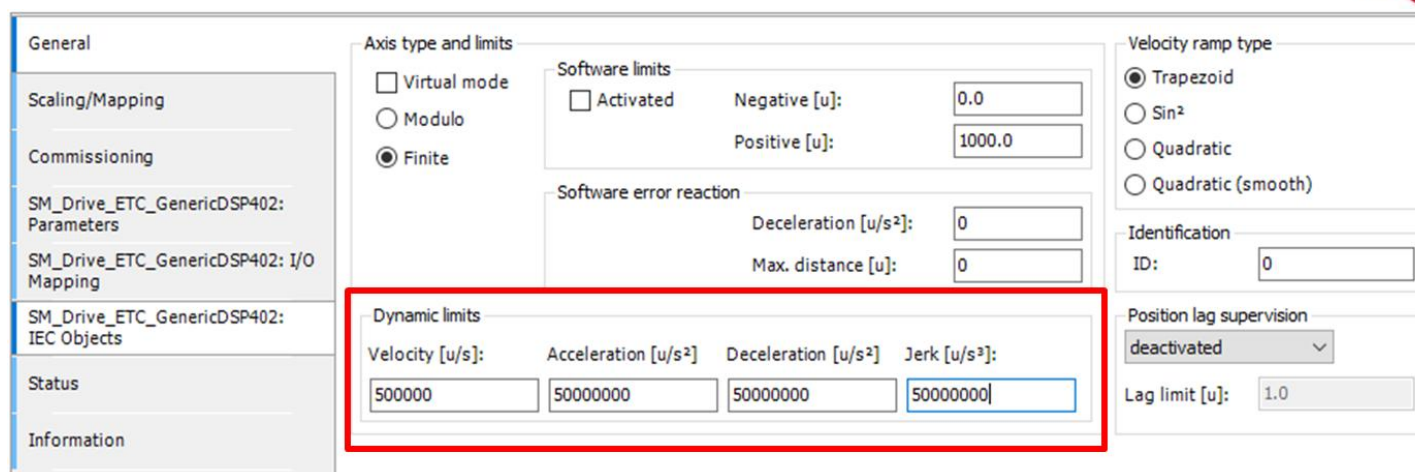
Сначала рассчитывается теоретически достижимое ускорение при разгоне до 3000 об/мин за 1 мс.

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./сек²

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 разгоняться до 3000 об/мин как минимум за 10 мс (50 000 000 ед./сек²). 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/сек²

Ускорение 20 мс = 25 000 000 ед/сек²

Скорость 3000 об/мин = 500 000 ед/сек

Скорость 300 об/мин = 50 000 ед/сек

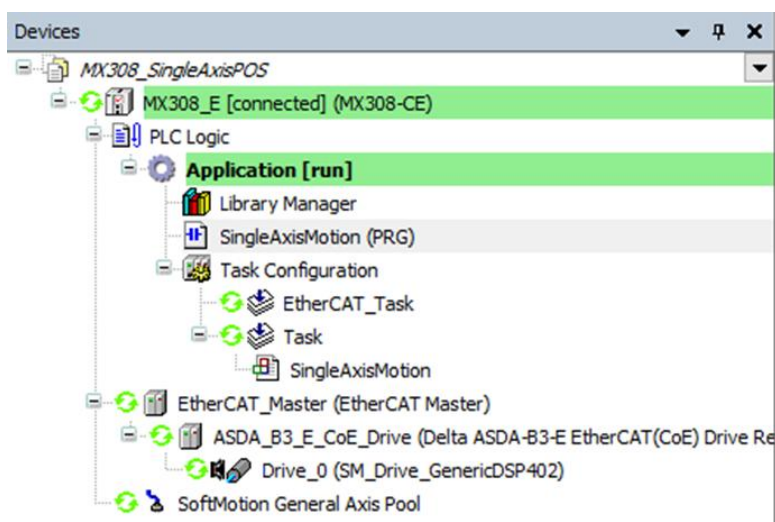
Скорость 15 об/мин = 2500 ед/сек

Перемещение на 10 мм = 10000 ед.

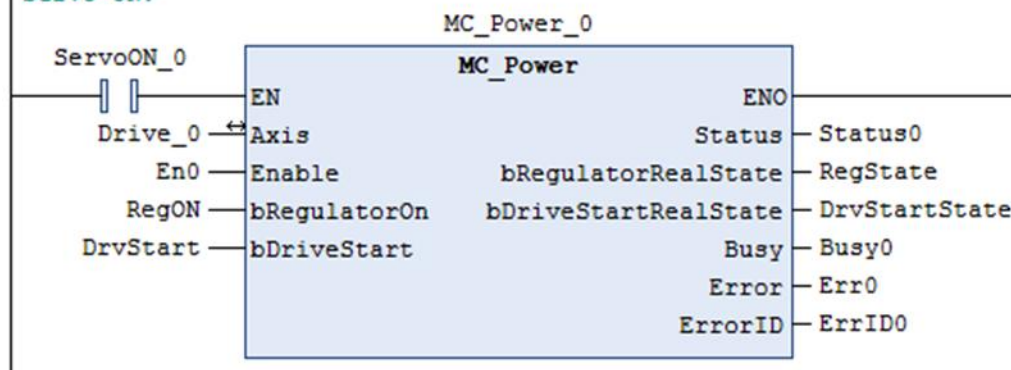
Перемещение на 20,456 мм = 20456 ед.

Перемещение на 300 мм = 300 000 ед.

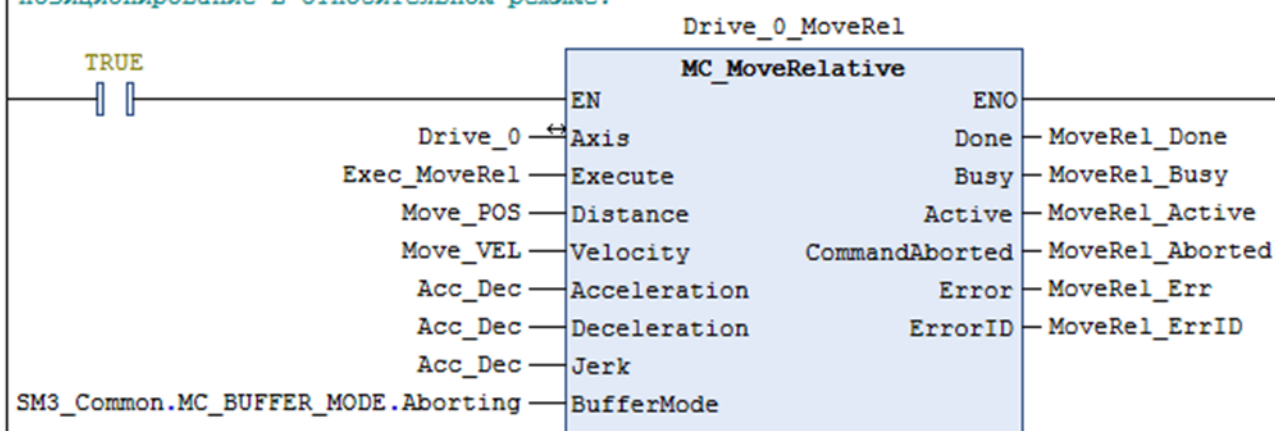
Ниже приводятся для примера ряд инструкций для одноосевого движения:



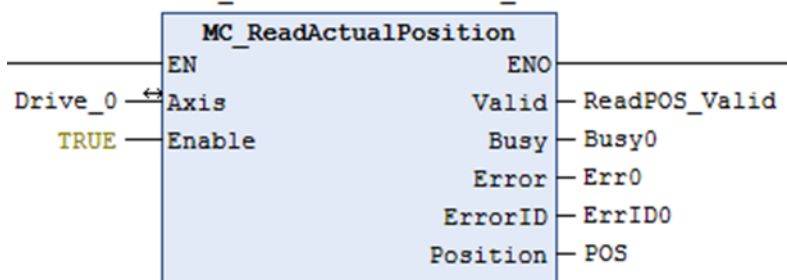
Servo-ON.



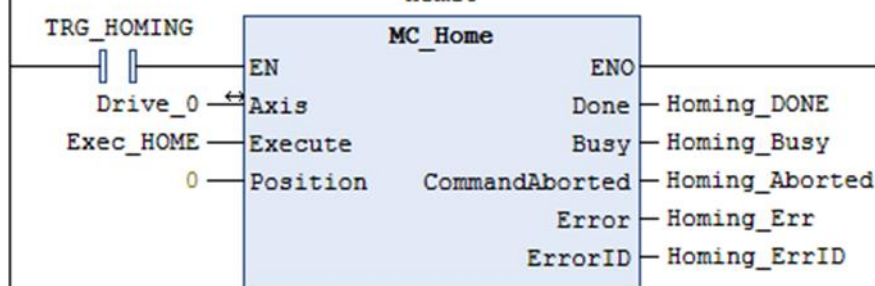
Позиционирование в относительном режиме.



MC_ReadActualPosition_0



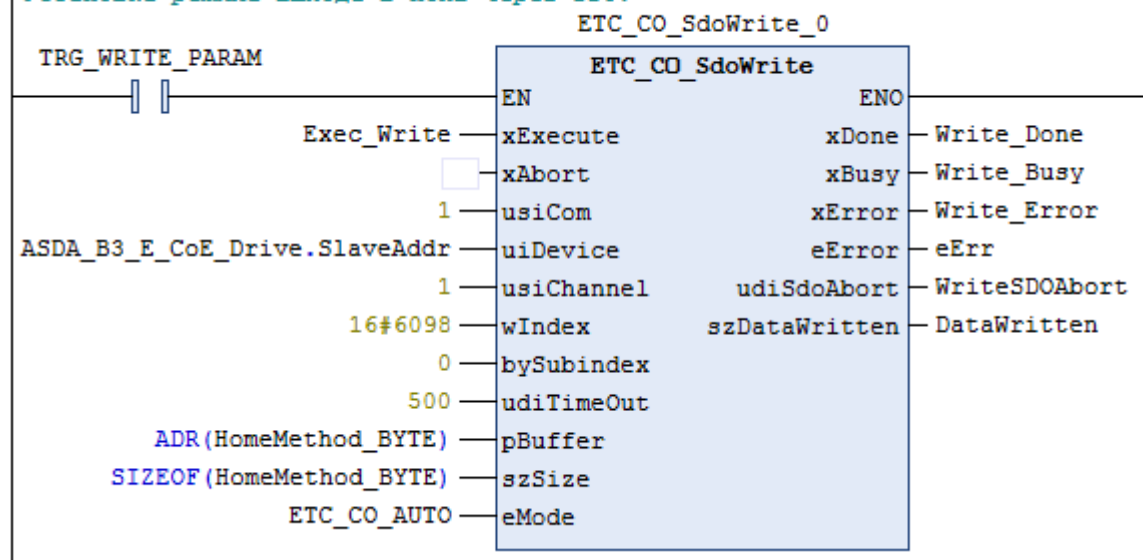
Home0



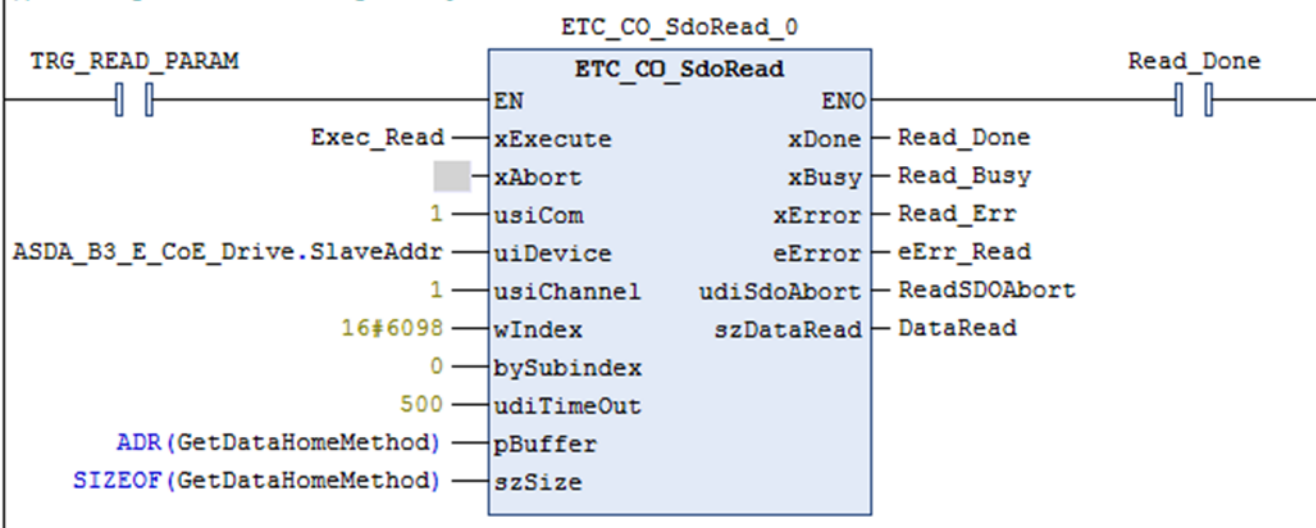
Для установки режима выхода в ноль, а также записи/чтению других параметров привода нужно использовать команды:

53 VAR HomeMethod_BYTE BYTE

Установка режима выхода в ноль через SDO.

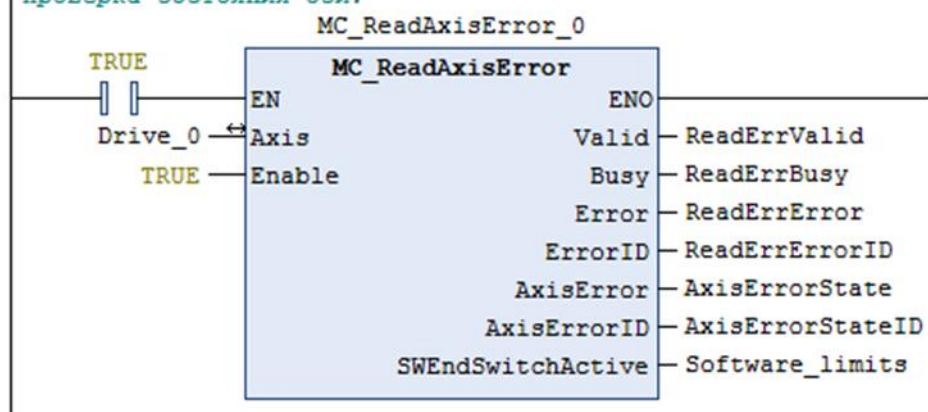


Чтение заданного режима выхода в ноль через SDO для проверки записи на предыдущем шаге. Данные принимаются в переменную GetDataHomeMethod типа BYTE.

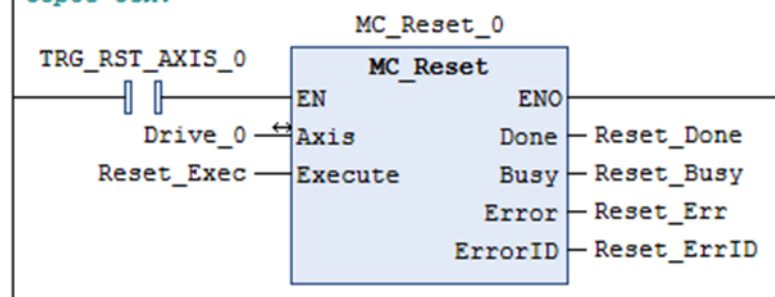


Для сброса оси и реинициализации привода используются команды:

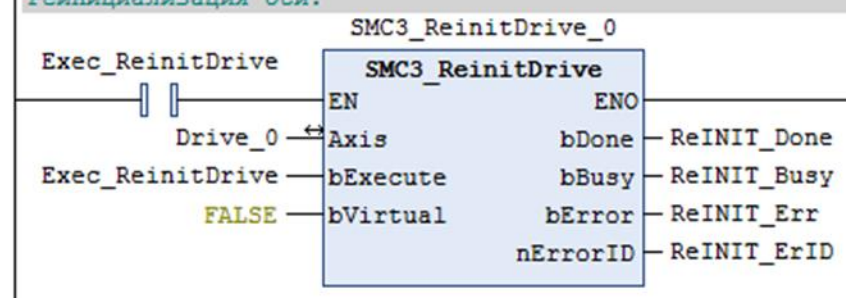
Проверка состояния оси.



Сброс оси.



Реинициализация оси.

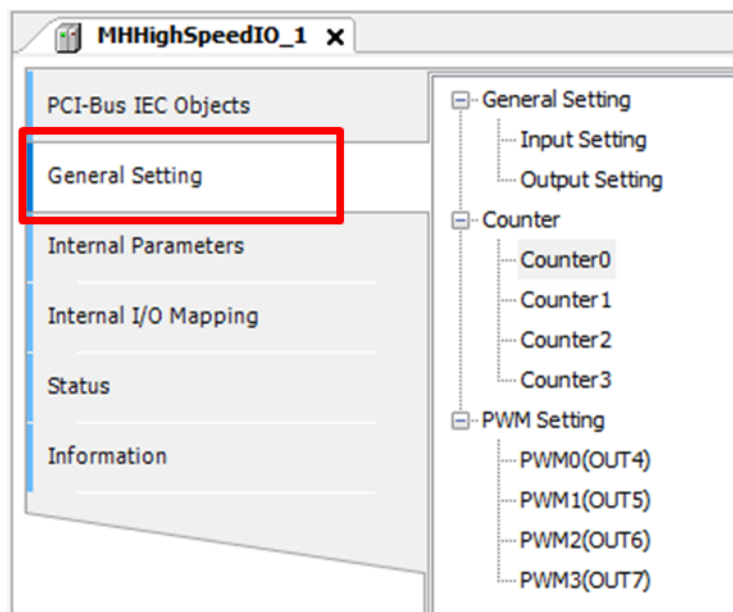


Работа с высокоскоростными счётчиками

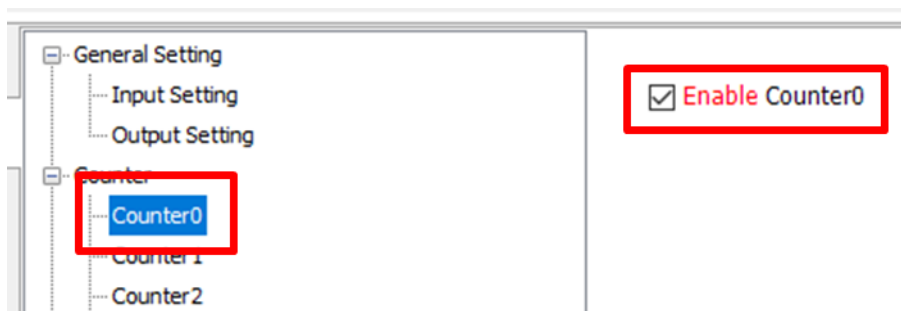
Контроллер имеет на борту 4 АВ-канала высокоскоростного счёта (входы 00-07). Далее описывается последовательность настройки высокоскоростных счётчиков.
(в проект должен быть добавлен заранее объект **MHHighSpeedIO**, см. данное Руководство).

Задание параметров работы высокоскоростных счётчиков

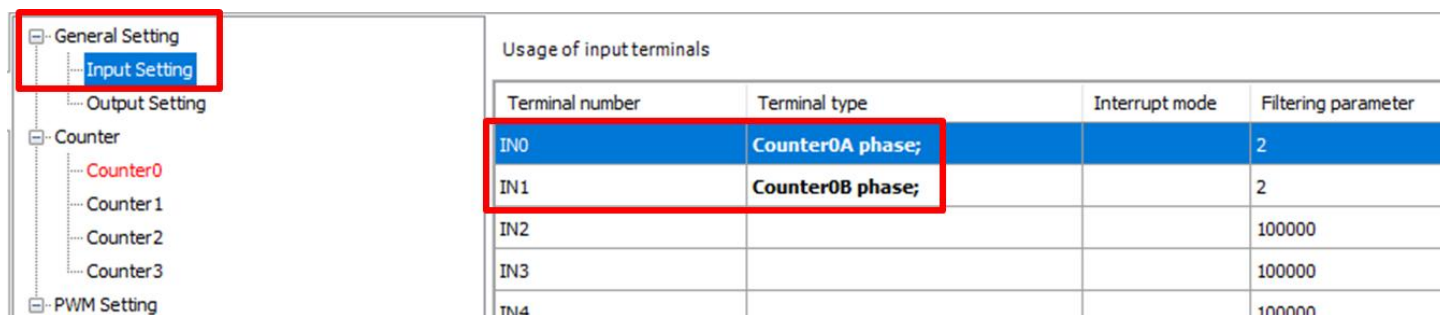
Щёлкните дважды на пункте "**MHHighSpeedIO**" в древе проекта и в открывшейся вкладке выберите раздел "**General Configuration**":



Выберите для примера пункт **Counter0** и поставьте флажок:



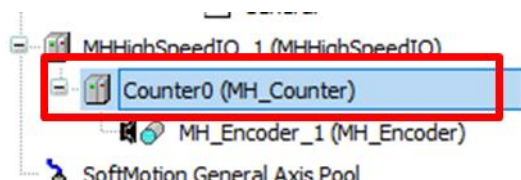
В пункте **Input Setting** появятся обозначения счётчика на входах 4 и 5.



Здесь же можно выставить входной фильтр:

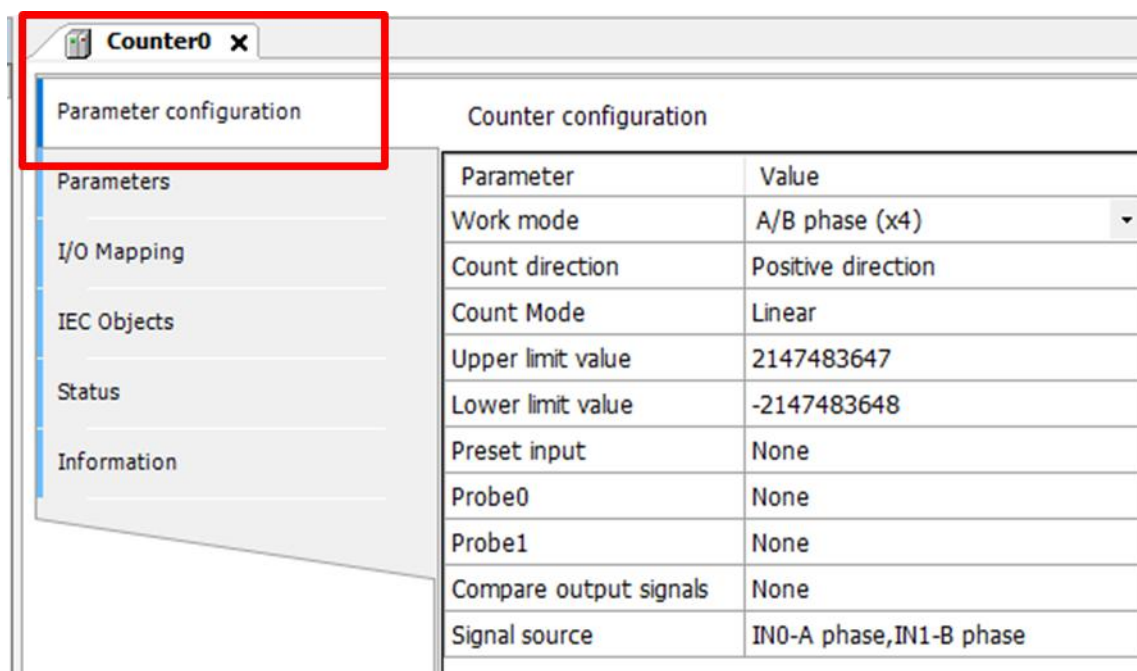
Terminal number	Terminal type	Interrupt mode	Filtering parameter	Notes
IN0	Counter0A phase;		2	Unit 100ns, high speed input (range 1-10000000)
IN1	Counter0B phase;		2	Unit 100ns, high speed input (range 1-10000000)

Также, в древе проекта под адаптером **MHHighSpeedIO** появится счётчик и привязанная к нему энкодерная ось:



Щёлкните на пункте **Counter0(MH_Counter)** два раза левой кнопкой мышки:

откроется вкладка с настройками счётчика. Выберите пункт **Parameter Configuration**:



В данной вкладке можно сделать настройки высокоскоростного счётчика:

Working mode – Рабочий режим (AB, импульс/направление и т.д.)

Count direction – Positive – счёт вверх, Negative – счёт вниз

Count mode – Linear – линейный счёт (до предела), Rotary – круговой счёт

Upper Limit Value – верхний предел счёта

Lower Limit Value – нижний предел счёта

Preset input – вход для аппаратного сброса счётчика (вход для инструкции LS_PresetValue)

Probe1 – вход 1 для инструкции LS_TouchProbe (привязка по названию энкодерной оси)

Probe2 – вход 2 для инструкции LS_TouchProbe (привязка по названию энкодерной оси)

Compare Output Signals – выход для инструкции LS_Compare (привязка по названию энкодерной оси)
Signal Source – назначение входов на ЦПУ для счётчика (можно изменить установленные по умолчанию)

Пункт **Parameters** служит для информативных целей в нём ничего менять не надо. Он сам модифицируется в зависимости от настроек в пункте **Parameter Configuration**.

В пункте **I/O Mapping** в колонке **Variable** можно задать названия переменным. Однако, данные переменные являются внутренними для команды **LS_Counter**, поэтому их можно использовать только после запуска данной команды в программе:

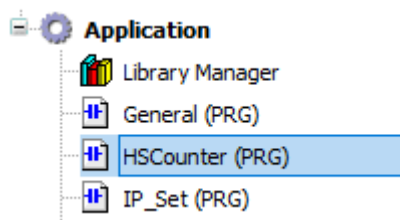
Counter0 X					
Parameter configuration					
Parameters					
I/O Mapping					
IEC Objects					
Status					
Information					
Find Filter Show all					
Variable Mapping Channel Address Type					
+ Counter0_Status Statusword %IW2 WORD					
+ Counter0_Value Counter value %ID2 DINT					
+ Touch Probe Status %IW6 WORD					
+ Preset status %IW7 WORD					
+ Compare status %IW8 WORD					
+ Touch probe0 posP %ID5 DINT					
+ Touch probe0 posN %ID6 DINT					
+ Touch probe0 timePL %ID7 UDINT					
+ Touch probe0 timePH %ID8 UDINT					
+ Touch probe0 timeNL %ID9 UDINT					
+ Touch probe0 timeNH %ID10 UDINT					
+ Touch probe1 posP %ID11 DINT					
+ Touch probe1 posN %ID12 DINT					
+ Touch probe1 timePL %ID13 UDINT					
+ Touch probe1 timePH %ID14 UDINT					
+ Touch probe1 timeNL %ID15 UDINT					
+ Touch probe1 timeNH %ID16 UDINT					
+ Current frequency %ID17 DINT					
+ Errorcode %IW36 UINT					
+ Controlword %QW2 WORD					
+ Touch probe command %QW3 WORD					
+ Preset command %QW4 WORD					
+ Preset Value %QD3 DINT					
+ Compare command %QW8 WORD					
+ Compare value %QD5 DINT					
+ Compare Step %QD6 UDINT					
+ Compare Number %QW14 UINT					
+ Output type %QW15 UINT					
+ Output Parameter %QD8 UDINT					

Для обращения к счётчикам в программе используйте команды:

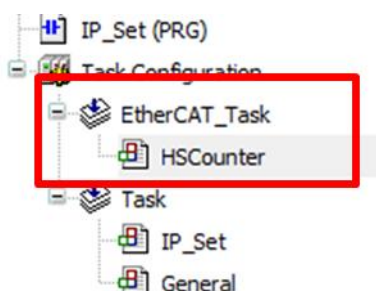
LS_Counter - для получения текущего значения счётчика

LS_PresetValue - для обнуления счётчика (с нулевым смещением, **fPresetValue = 0**).

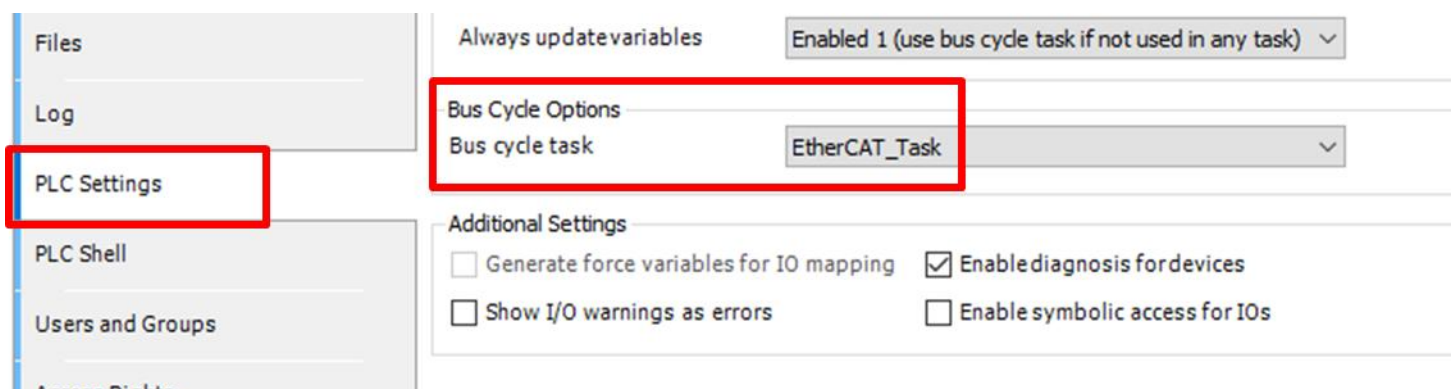
Для примера создадим программный блок (POU) **HSCounter**, где будет использоваться высокоскоростной счётчик.



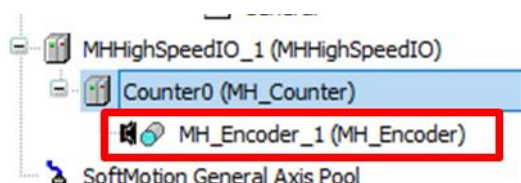
Программный модуль (POU, в нашем примере HSCounter), где будет использоваться команды работы с высокоскоростным счётчиком, должен быть привязан в Менеджере Задач (**Task Configuration**) к основной задаче (приоритет 0) **EtherCAT_Task**:

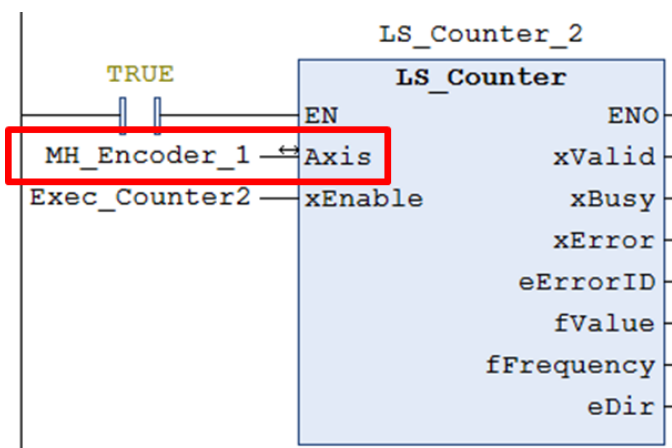


Также, во вкладке Device в пункте PLC Settings нужно выбрать в качестве такта шины (Bus cycle task) привязать основную задачу (приоритет 0) **HSIOTask**:

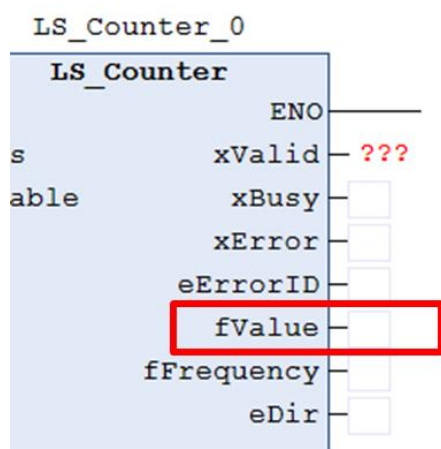


Для обращения к счётчику в командах необходимо использовать имя энкодерной оси в древе проекта (в нашем проекте **MH_Encoder_1**):

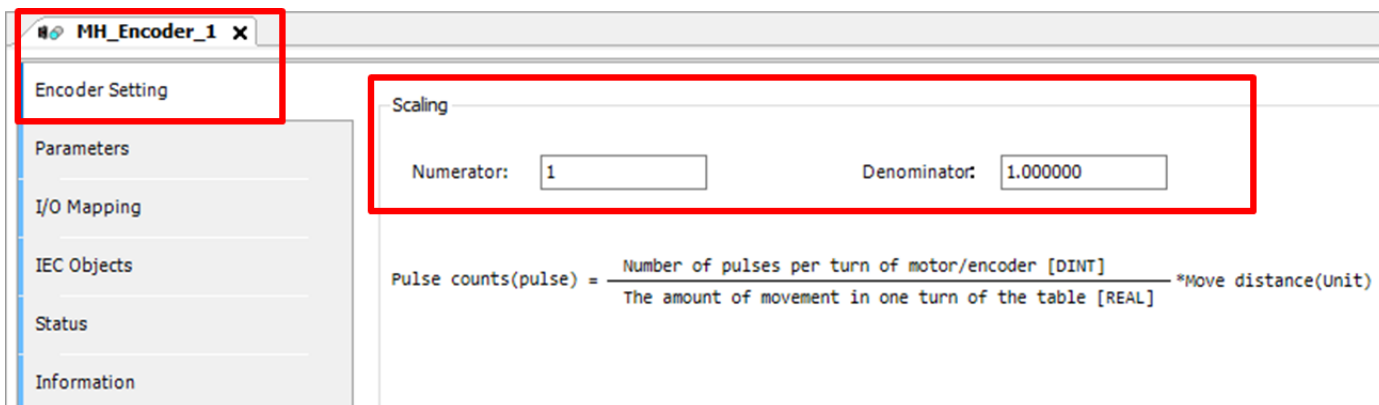




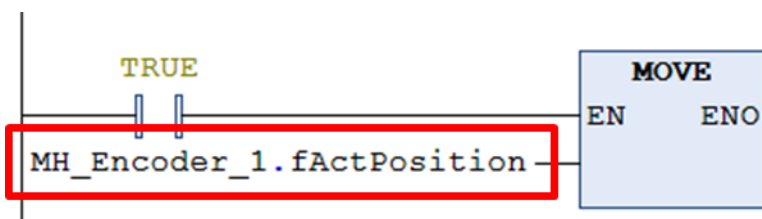
Для получения значения счётчика нужно использовать ножку инструкции **fValue**:



Если в настройках энкодерной оси коэффициенты **Numerator** и **Denominator** равны 1, то позиция оси и значение счётчика будут прямо соответствовать количеству физически полученных импульсов на счётном входе:

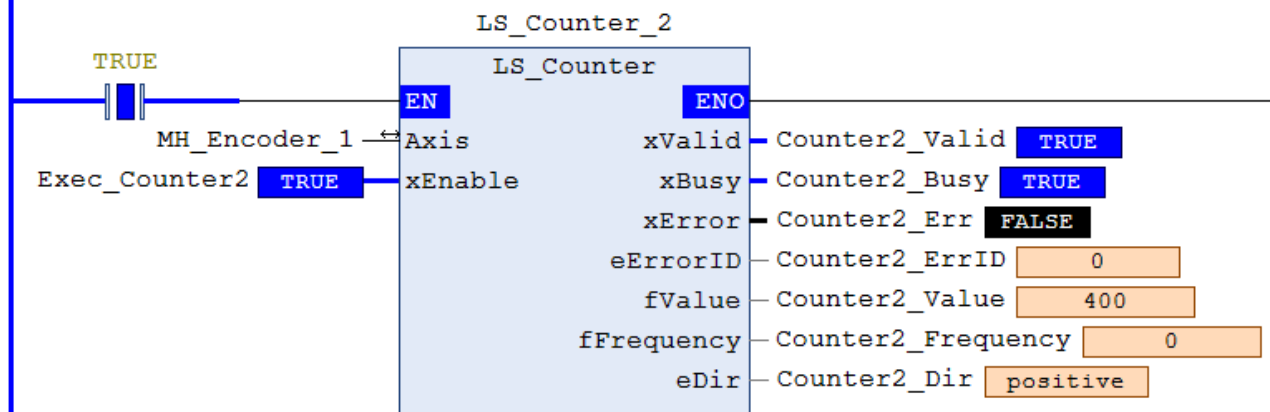


Мониторинговые переменные выглядят так:

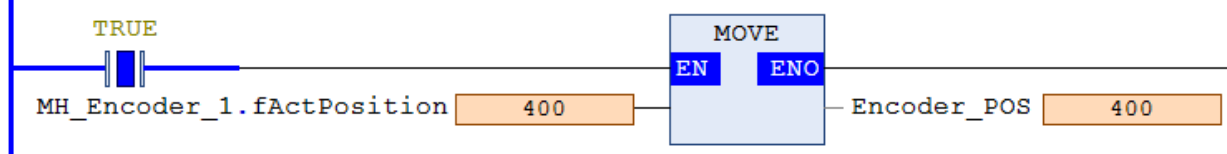


Пример инструкции высокоскоростного счёта:

Инструкция высокоскоростного счёта.
Для привязки используется имя энкодерной оси.
Необходимо учитывать числитель/знаменатель оси!
Их значения влияют на выход fValue.

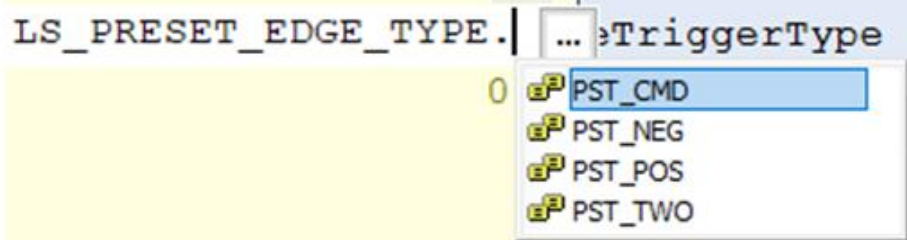
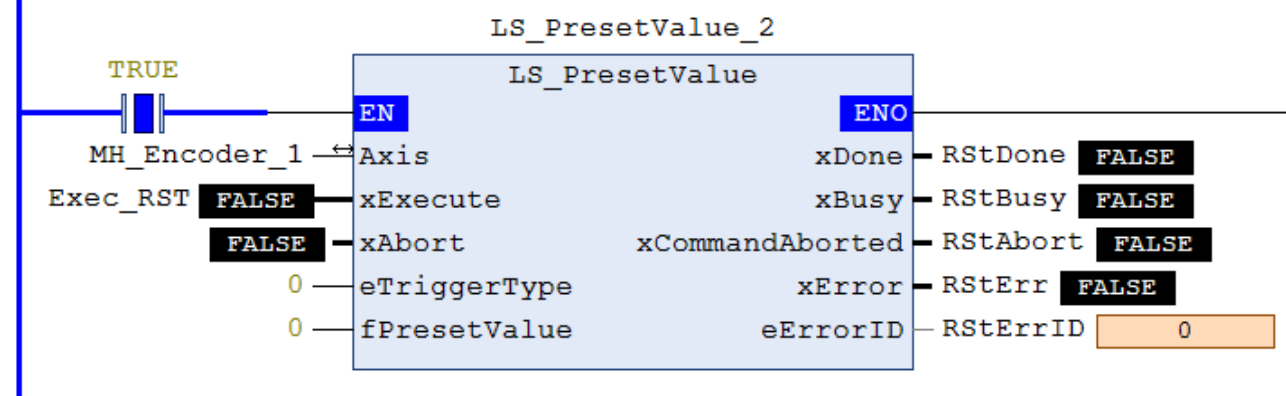


Текущее положение оси через мониторинговую переменную:



Пример инструкции сброса счётчика на ноль из программы:

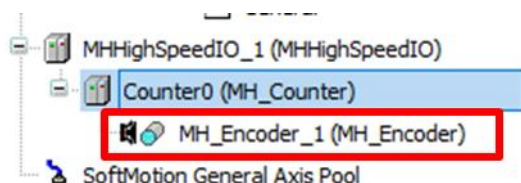
Сброс счётчика на 0. Для привязки используется имя энкодерной оси.
На ноже eTriggerType стоит 0, что означает сброс из программы.
Если на этой ноже поставить 1, 2 или 3, то будет использоваться вход на ЦПУ, назначенный в настройках счётчика.



eTriggerType: структура (перечисление) **LS_PRESET_EDGE_TYPE**.****

Поля структуры: PST_CMD – сброс из программы по ножке **xExecute**, PST_NEG – по заднему фронту, PST_POS – по переднему фронту, PST_TWO – по обоим фронтам. Вход назначается в настройках каждого счётчика отдельно, пункт **Preset input**

Как было показано выше при назначении высокоскоростного счётчика в древе проекта к нему автоматически привязывается энкодерная ось:

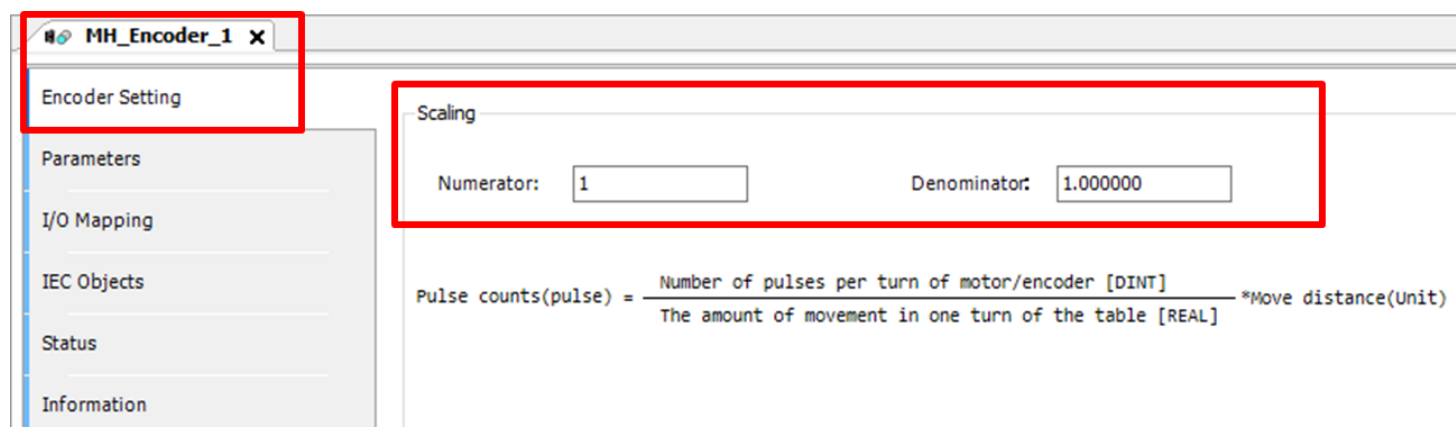


Энкодерная ось может использоваться в качестве Мастер оси в стандартных инструкциях управления движением типа MC_Gear и т.п.

Текущее положение можно получить из параметра **fActPosition** оси структуры оси:

MH_Encoder_1.fActPosition

Настройки коэффициента редукции осуществляется в пункте **Encoder Setting**:



Если коэффициенты **Numerator** и **Denominator** равны 1, то позиция оси и значение счётчика будут прямо соответствовать количеству физически полученных импульсов на счётном входе:

MH_Encoder_1.fActPosition = HSCounters.LS_Counter_2.fValue = количество полученных импульсов

Например, если подключить стандартный штурвальный энкодер на 100 импульсов/оборот в режиме 4хА/В, то на 1 оборот будет принято 400 импульсов, значение счётчика и позиция оси также будет 400:

Watch 1			
Expression	Application	Type	Value
 HSCounter.LS_Counter_2.fValue	Controller_1.Application	LREAL	400
 MH_Encoder_1.fActPosition	Controller_1.Application	LREAL	400

Если внести редукцию, то принятое количество импульсов и позиция оси/значение счётчика будут отличаться. Коэффициент может быть только повышающим. Формула сводится к следующему:

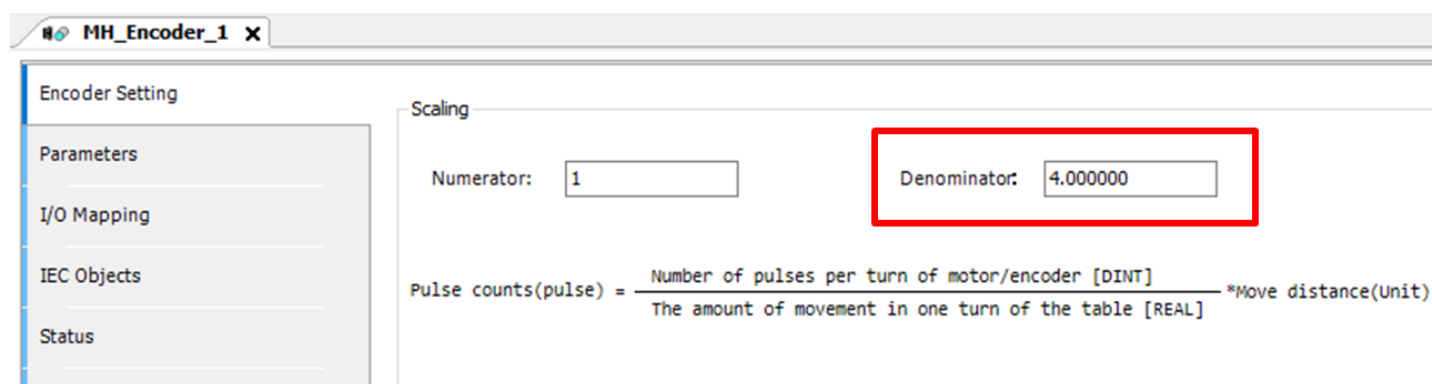
Позиция оси/значение счётчика = входные импульсы * Denominator/Nominator

MH_Encoder_1.fActPosition = HSCounters.LS_Counter_2.fValue = количество полученных импульсов * Denominator/Nominator

Внимание!

Denominator/Nominator >= 1 (больше или равно единицы)

Например, если сделать Denominator = 4 (Nominator=1), то позиция оси и значение счётчика будет в 4 раза больше, чем физически принятых импульсов на счётном входе.



Например, если подключить стандартный штурвальный энкодер на 100 импульсов/оборот в режиме 4хА/В, то на 1 оборот будет принято 400 импульсов, а значение счётчика и позиция оси будут 1600:

Watch 1			
Expression	Application	Type	Value
 HSCounter.LS_Counter_2.fValue	Controller_1.Application	LREAL	1600
 MH_Encoder_1.fActPosition	Controller_1.Application	LREAL	1600

Принятое количество импульсов можно пересчитать в линейное перемещение по следующей формуле:

Перемещение (user units) = количество принятых импульсов * линейное перемещение (user units) на 1 оборот энкодера / количество импульсов на 1 оборот энкодера

Предположим у нас 1000 импульсов на 1 оборот энкодера, механизм проходит 10 мм (user units) на 1 оборот энкодера. Контроллер принял 4000 импульсов. Следовательно пройденное расстояние механизмом будет:

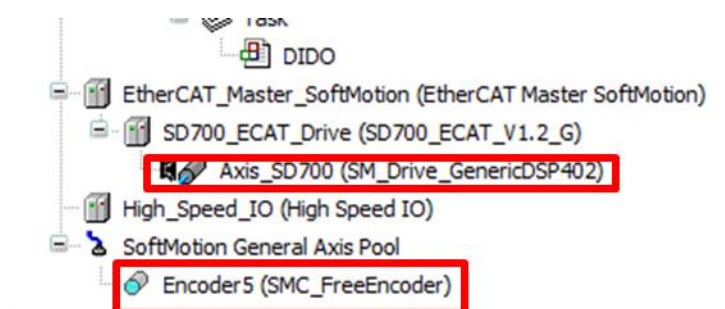
$S \text{ (user units)} = 4000 * 10 / 1000 = 40 \text{ мм}$

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 на ЦПУ. Процедура организации высокоскоростных счётчиков и привязки энкодерных осей рассмотрены в параграфе «Работа с высокоскоростными счётчиками», 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.

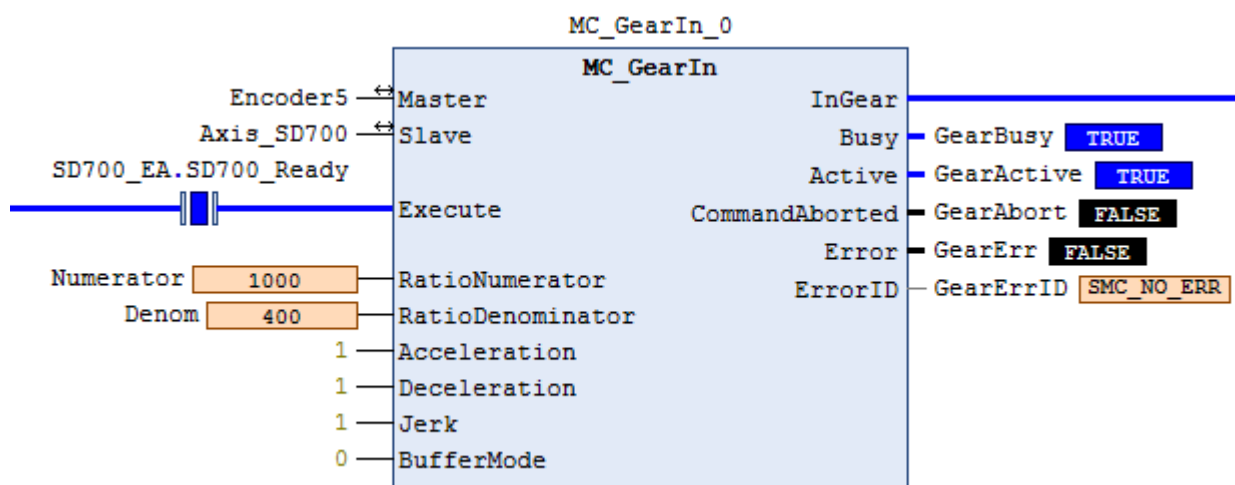
Counter0 x									
Parameter configuration	Counter configuration								
Parameters	<table border="1"> <thead> <tr> <th>Parameter</th><th>Value</th></tr> </thead> <tbody> <tr> <td>Work mode</td><td>A/B phase (x4)</td></tr> <tr> <td>Count direction</td><td>Positive direction</td></tr> <tr> <td>Count Mode</td><td>Linear</td></tr> </tbody> </table>	Parameter	Value	Work mode	A/B phase (x4)	Count direction	Positive direction	Count Mode	Linear
Parameter	Value								
Work mode	A/B phase (x4)								
Count direction	Positive direction								
Count Mode	Linear								
I/O Mapping									
IEC Objects									

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	131072	increments <=> motor turns	1
☐ Linear	1	motor turns <=> gear output turns	1
	1	gear output turns <=> units in application	1000

Перед началом работы на природ необходимо подать команду Servo-ON (блок MC_Power).

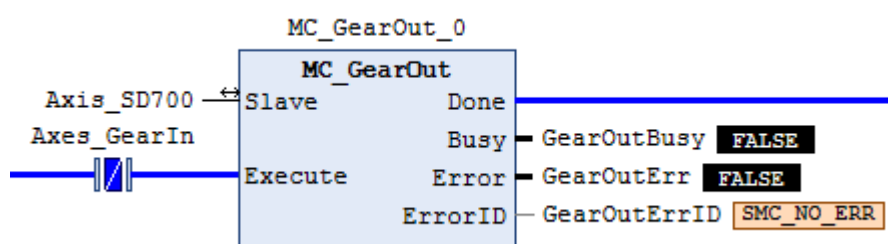
Связка Мастер ось – Ведомая ось устанавливается через команду MC_GearIn:



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).

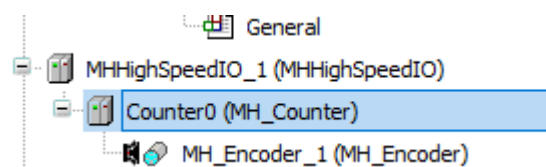
Расцепление осей осуществляется командой MC_GearOut:



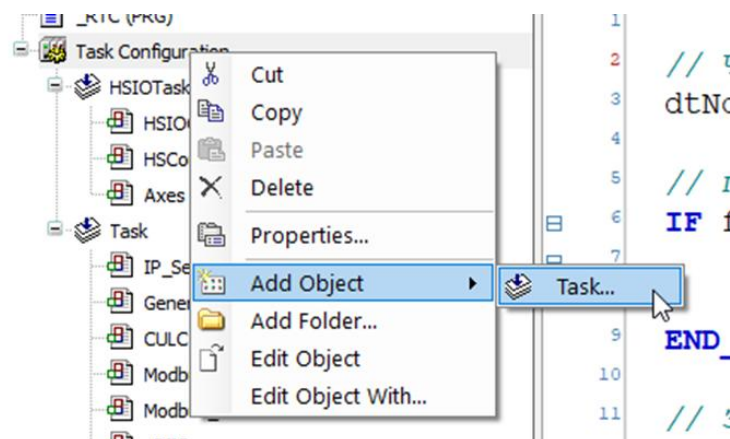
Высокоскоростное сравнение

В ряде задач требуется выполнить сравнение текущего значения высокоскоростного счётчика с заданным порогом и исполнить в режиме прерывания определённый код или включить выход на контроллере. Для реализации подобной процедуры необходимо выполнить ряд действий.

Для реализации высокоскоростного сравнения в древо проекта контроллера должен быть добавлен адаптер входов-выходов на ЦПУ – **MHHighSpeedIO** (см. соответствующую Главу данного Руководства). Это необходимо для организации высокоскоростного счётчика, текущее значение оси которого и будет сравниваться с заданным порогом. Создайте счётчик **Counter0** (см. Работа с высокоскоростными счётчиками, стр. 91 настоящего Руководства).

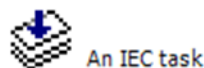


В начале необходимо создать Задачу (Task) обработки прерывания. Для этого встаньте в древе проекта на пункт **Task Configuration** и правой кнопкой мышки вызовите меню, в котором выберите пункт **Add Object – Task**:



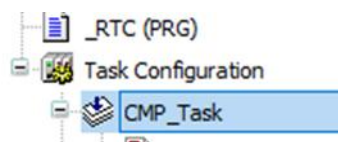
Дайте название Задаче, например **CMP_Task**.

Add Task

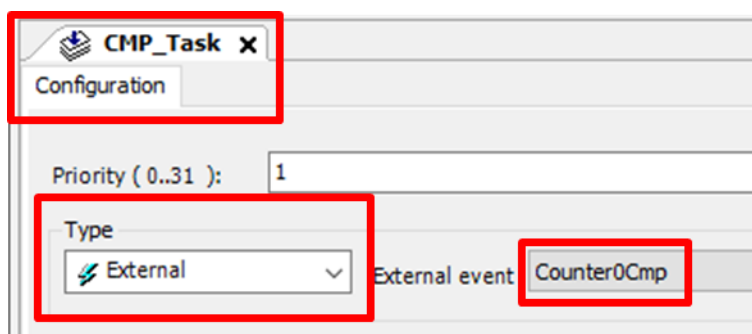


Name
CMP_Task

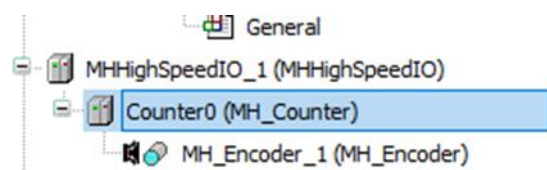
В древе проекта появится Задача **CMP_Task**:



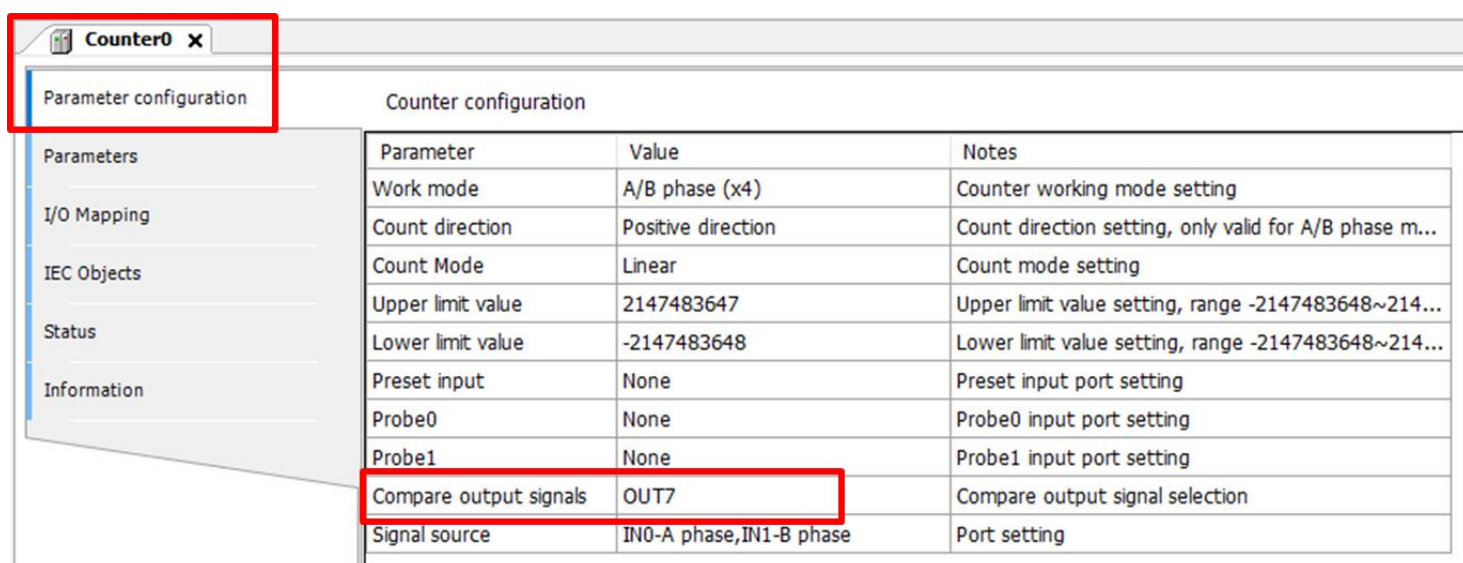
Щёлкните дважды левой кнопкой мышки на пункте **CMP_Task**, откроется вкладка с настройками параметров Задачи. Выберите тип задачи **External** и назначьте счётчик, в нашем примере **Counter0Cmp**:



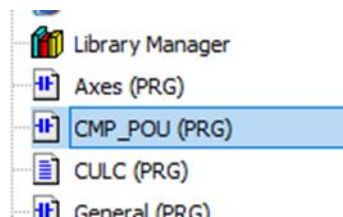
Далее щёлкните дважды левой кнопкой мышки на пункте **Counter0** в древе проекта:



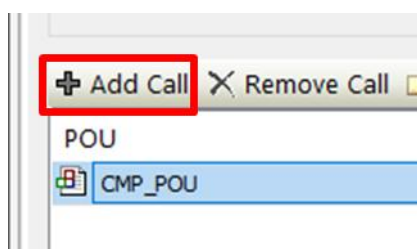
Откроется вкладка с настройками счётчика, в которой необходимо выбрать раздел **Parameter Configuration** и в параметре **Compare Output Signals** выбрать выход, который будет включаться при наступлении события прерывания по достижению порога высокоскоростного счёта. На рисунке ниже выбран выход 06 (**OUT6**):



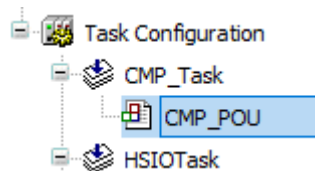
Далее создайте программный блок, например **CMP_POU**:



и привяжите его к Задаче **CMP_Task**:

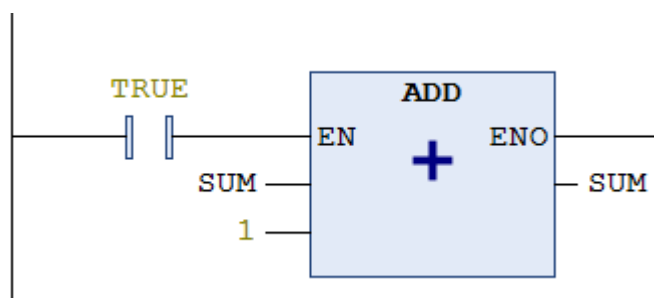


В древе проекта появится пункт:



Далее в программном блоке CMP_POU для целей демонстрации можно создать простой код, который будет исполняться только по наступлению события достижения порога высокоскоростного счёта:

Scope	Name	Address	Data type
VAR	SUM		INT

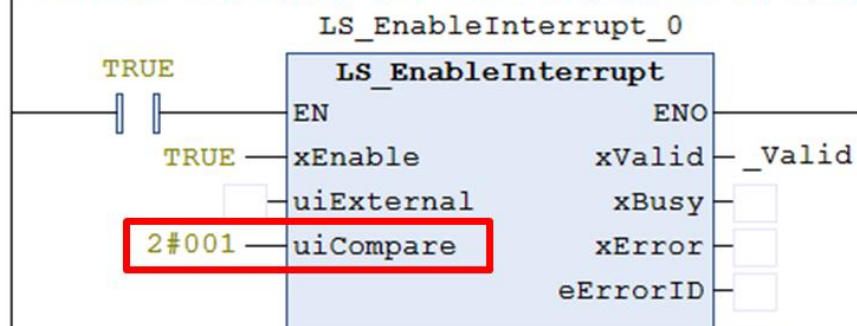


(если бы данная команда стояла в циклической Задаче, то исполнялась бы в каждом скане, а так только по наступлению события).

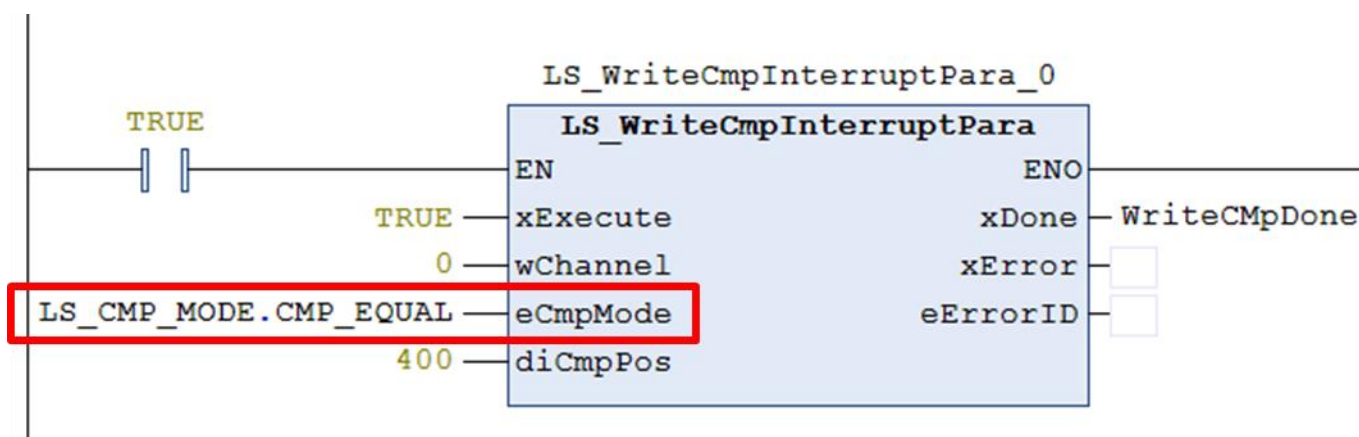
Прерывание необходимо зарегистрировать в команде **LS_EnableInterrupt** в двоичном коде по номеру счётчика:

Разрешение прерываний. Задаётся в двоичном коде.

На ножке uiCompare включено прерывание по сравнению на счётчике Counter0.

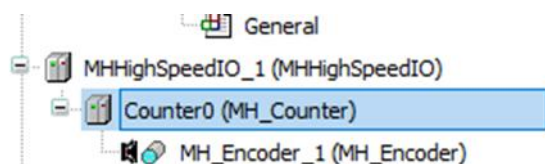


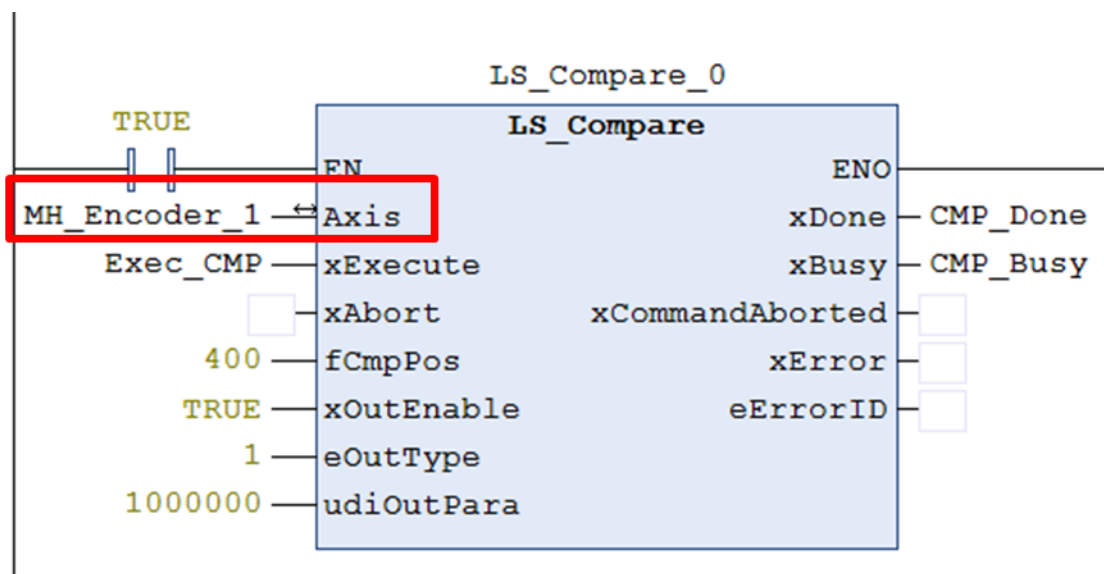
Также, в программе должна стоять команда **LS_WriteCmpInterruptPara** с параметром **eCmpMode** равным **LS_CMP_MODE.CMP_EQUAL**:



Обе инструкции должны быть постоянно включены (**TRUE** на ножках **EN** и **xExecute**).

Далее в любой циклической задаче необходимо поставить команду **LS_Compare** и привязать по **имени оси** счётчик:





Описание основных ножек ФБ:

Axis – имя энкодерной оси, привязанной к счетчику в древе проекта

xExecute – запускает ФБ (необходимо взводить заново каждый раз после срабатывания ФБ)

fCmpPos – порог для сравнения. Количество импульсов в user units LREAL

eOutType – режим сравнения, поставить 1 (выход включен на установленное время по наступлению события)

xOutEnable – разрешение установки выхода при достижении порога, поставить TRUE

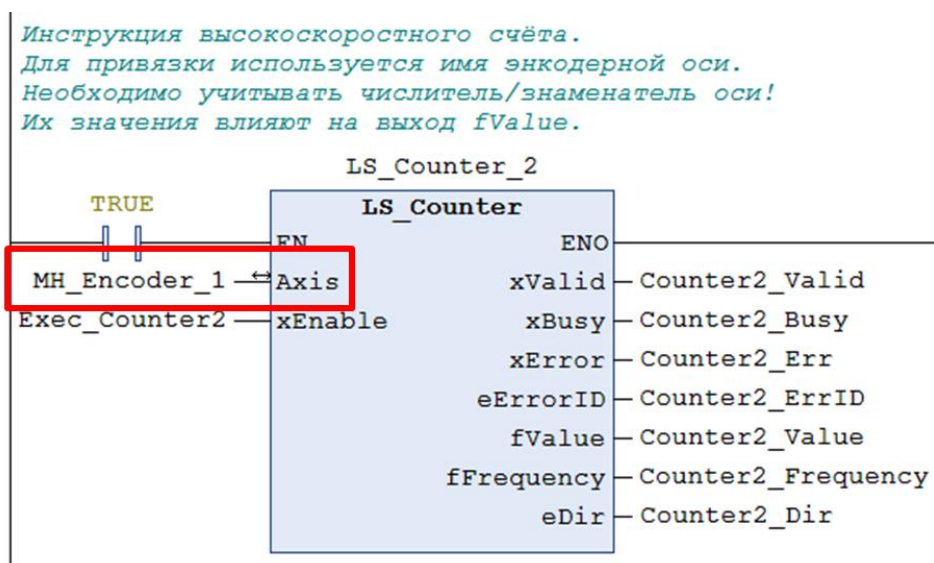
udiOutPara – время удержания выхода при срабатывании прерывания, мкс (1000000 = 1 секунда)

CMP_Done – флаг окончания сравнения

CMP_Busy – флаг готовности ФБ к сравнению

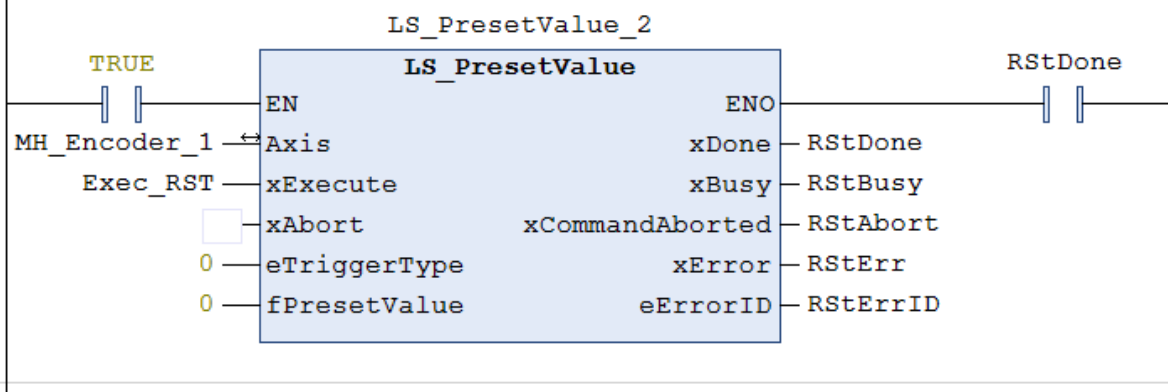
При работе с командой **LS_Compare** необходимо учитывать, что она срабатывает по фронту сигнала на ножке **xExecute**, загорается флаг готовности **xBusy** и команда готова к высокоскоростному сравнению. После выполнения сравнения флаг **xBusy** гаснет и загорается флаг **xDone**. Для осуществления следующей операции сравнения необходимо погасить и снова взвести ножку **xExecute**. Также, в программе у Вас должна быть включена инструкция высокоскоростного счёта для соответствующего счётчика **LS_Counter**. Сброс счётчика осуществляется командой **LS_PresetValue** со значением 0.

Команда активации счётчика:



Команда сброса счётчика:

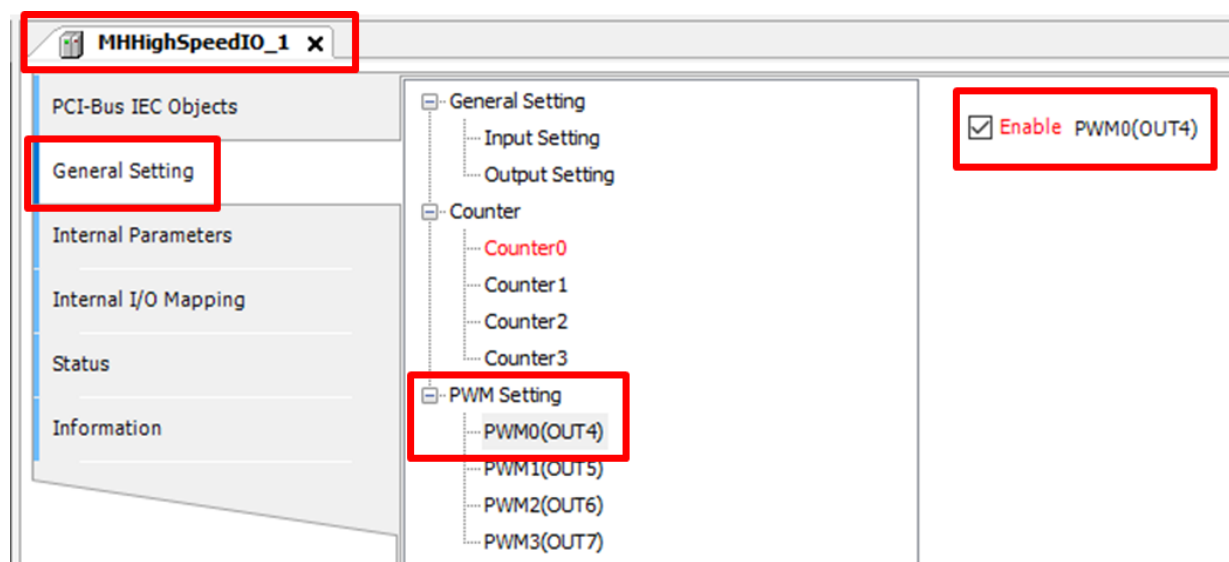
Сброс счётчика на 0. Для привязки используется имя энкодерной оси.
На ножке eTriggerType стоит 0, что означает сброс из программы.
Если на этой ножке поставить 1, 2 или 3, то будет использоваться вход на ЦПУ, назначенный в настройках счётчика.



Обратите внимание, что во всех инструкциях привязка идёт к имени оси, а не к номеру счётчика.

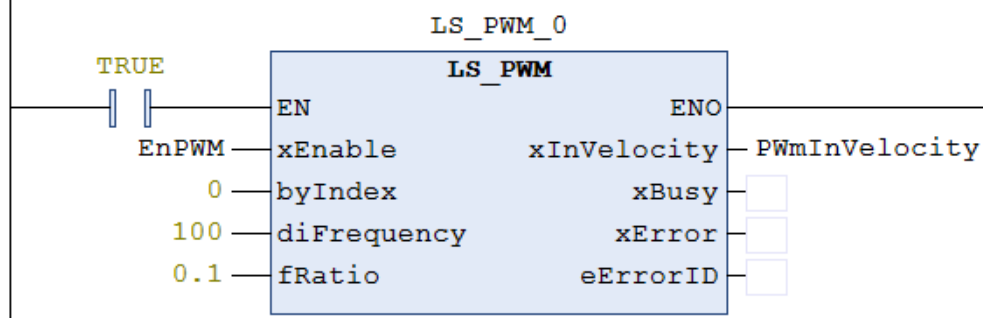
Работа в режиме ШИМ импульсов

Контроллеры типа МН1048 имеют аппаратную возможность формировать ШИМ импульсы на выходах 4 - 7. Для организации данного функционала в проекте должен быть адаптер входов-выходов **MHHighSpeedIO**. Щёлкните дважды левой кнопкой мышки на пункте **MHHighSpeedIO** и в открывшейся вкладке выберите пункт **General Setting – PWM Setting – PWM0(OUT4)**. Поставьте флажок **Enable PWM0(OUT4)**:



Добавьте в программу команду **LS_PWM**:

*Частоту можно менять только при снятом Enable
Герцы не ниже 100
скважность REAL 0.0 – 1.0 (1 – выход включен всегда)*



Команда **LS_PWM** должна обязательно стоять в программном блоке, привязанном к основной Задаче **HSIOTask**.

Описание ножек ФБ:

xEnable – запуск ФБ. ШИМ выдаётся пока есть TRUE

byIndex – номер выхода, 0 – OUT14, 1 – OUT15

diFrequency – частота ШИМ импульсов. Диапазон 100 Гц – 200 кГц

fRatio – скважность 0-1 REAL. 0 – выход полностью закрыт, 1 – выход полностью открыт

xInVelocity – флаг, что импульсы выдаются

xBusy – флаг, что ФБ в работе

xError – флаг ошибки

eErrorID – код ошибки

Чтение и установка часов реального времени

Контроллеры семейства МН1000 имеют встроенные часы реального времени (RTC). Для поддержки работы часов используется стандартная батарейка CR2032, установленная в слоте (батхолдере) в корпусе контроллера. В составе базового пэкиджа с описанием устройства

 OD M Series Package 1.0.3.11

ставится библиотека **OD Basic**:

 OD = OD Basic, 1.0.0.1 (Optimus Drive) OD 1.0.0.1

В данной библиотеке содержится Функциональный Блок **OD_RTC**, в котором есть два метода **Get** и **Set** для обеспечения доступа к функционалу данного ФБ. Метод **Get** позволяет получить текущее значение часов реального времени контроллера в формате DT, а метод **Set** позволяет установить часы реального времени из программы контроллера также в формате DT.

 OD Basic, 1.0.0.1 (Optimus Drive)

- DTs n Vars
- FBs
- FNs
- Images
- Objects
 - OD_RTC
 - Get
 - Set

Для конвертации формата DT в тип UINT библиотека **OD Basic** содержит три функции **DTtostDT**, **stDTtoDT** и **uiDTtoDT**:

 OD Basic, 1.0.0.1 (Optimus Drive)

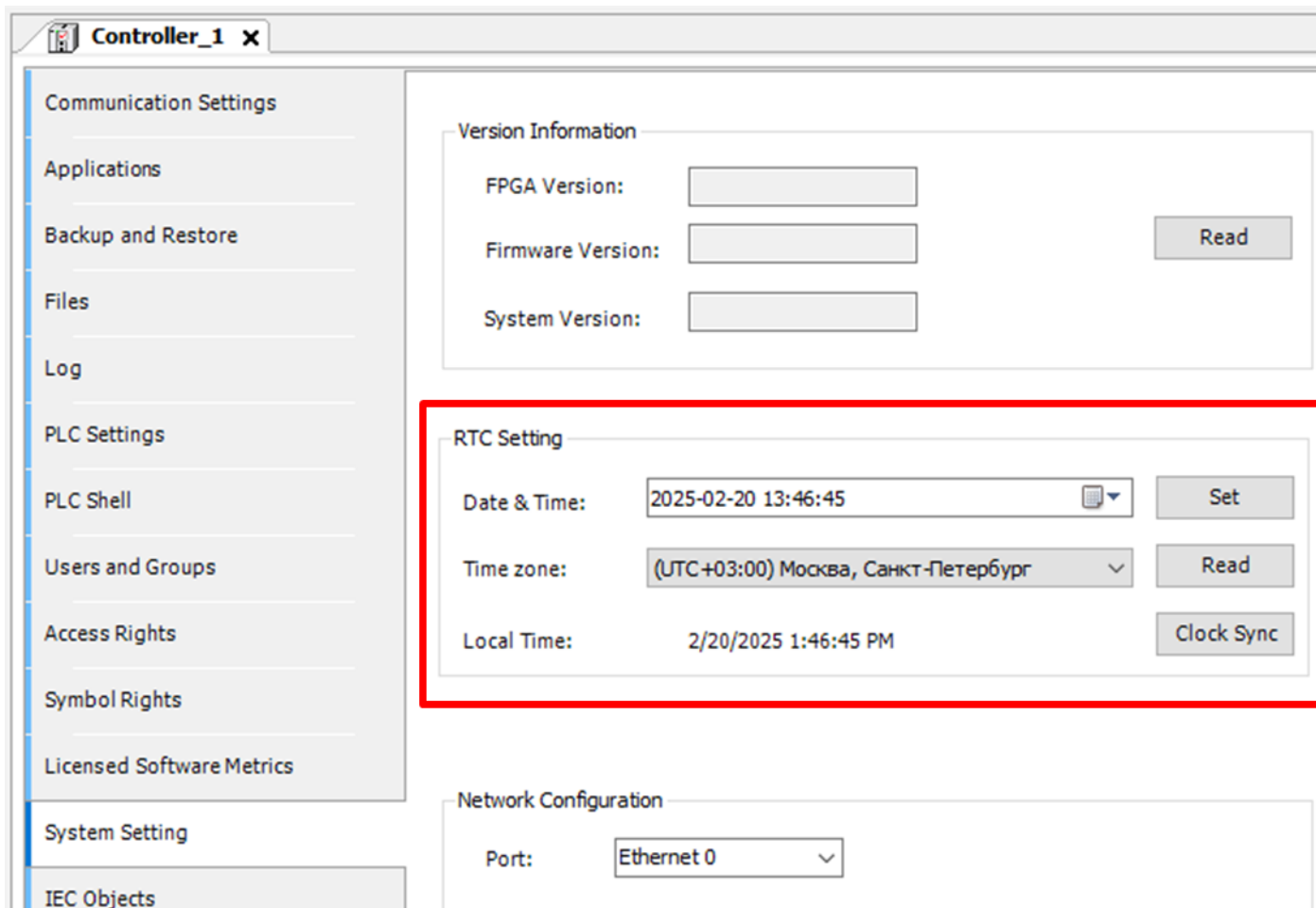
- DTs n Vars
- FBs
- FNs
 - DT
 - DTtostDT
 - DT_TO_STRF
 - FSTRING_TO_DT
 - MINUTE_TO_TIME
 - stDTtoDT
 - uiDTtoDT
- Logic

DTtostDT – функция конвертации DT формата даты, времени в структуру с элементами типа UINT

stDTtoDT – функция конвертации структуры даты, времени с элементами типа UINT в значение в DT формате

uiDTtoDT – функция конвертации UINT значений даты, времени в DT формат

Часы реального времени можно выставить также из среды программирования. Для этого необходимо открыть вкладку **Device** пункт **System Settings**:



Выставить нужные Дату и Время, а затем кнопкой **Set** записать в контроллер.
(связь с контроллером должна быть установлена заранее)

Кнопкой **Read** можно прочесть текущее значение часов реального времени контроллера.

Чтение часов реального времени контроллера методом OD_RTC.Get

В среде программирования CODESYS методы, как и функции, можно вызывать только на языке ST, поэтому вызов метода будет приведён на этом языке:

```
// Чтение часов реального времени (RTC)
dtNow[DT#2025-2-21-20:6:7] := fbOD_RTC.Get(); // Постоянное чтение часов реального времени UTC+3
```

```
PROGRAM _RTC
VAR
    dtNow:          DT; //переменная для отображения считанной текущей Даты/Времени
    fbOD_RTC:       OD_OD_RTC; //объявление экземпляра ФБ OD_RTC
END_VAR
```

```
// Чтение часов реального времени (RTC)
dtNow := fbOD_RTC.Get(); // Постоянное чтение часов реального времени UTC+3: Moscow Time Zone
```

Запись часов реального времени контроллера методом OD_RTC.Set

```
// Запись часов реального времени (RTC)
IF xSetDT[FALSE] THEN
    fbOD_RTC.Set(dtSetDT[DT#2025-2-21-11:3:0], 180); // Запись. UTC+3: Moscow Time Zone UTC+3 :
    xSetDT[FALSE] := FALSE;
END_IF[RETURN]
```

```
PROGRAM _RTC
VAR
    dtSetDT:          DT; //переменная для хранения Даты/Времени для записи в контроллер
    fbOD_RTC:       OD_OD_RTC; //объявление экземпляра ФБ OD_RTC
    xSetDT:         BOOL; //Триггер записи времени в контроллер
END_VAR
```

```
// Запись часов реального времени (RTC)
IF xSetDT THEN
    fbOD_RTC.Set(dtSetDT, 180); // Запись. UTC+3: Moscow Time Zone UTC+3 == iBias = 3*60 = 180
    xSetDT := FALSE;
END_IF
```

Примечание:

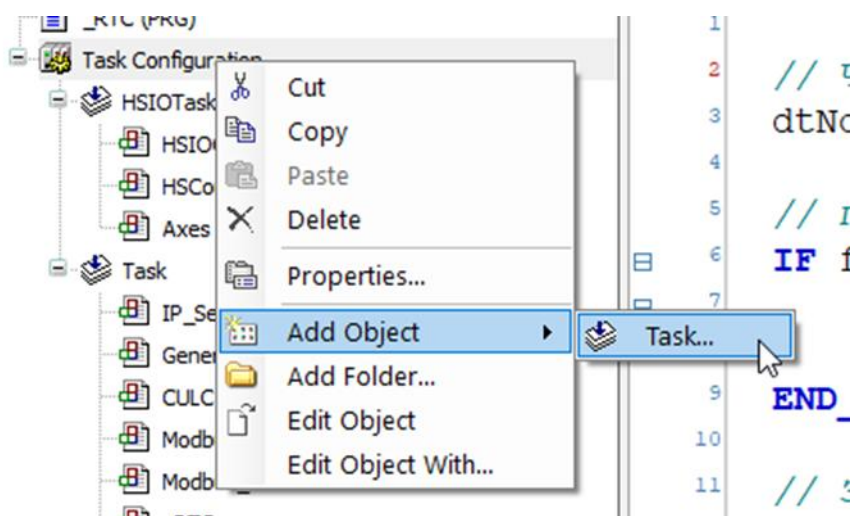
Минимальный год, который можно выставить – 2021

Прерывание по сигналу на входе контроллера

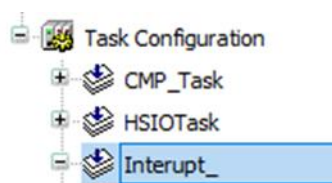
Контроллеры типа MN1048 имеют возможность обрабатывать назначенные программные блоки по фронту на дискретном входе контроллера. Прерывания могут быть назначены на входы 00 – 05 контроллера (всего 6 прерываний). Можно использовать передний, задний или оба фронта.

Для назначения прерываний по входам в древо проекта контроллера должен быть добавлен адаптер входов-выходов на ЦПУ – **MNHighSpeedIO** (см. соответствующую Главу данного Руководства).

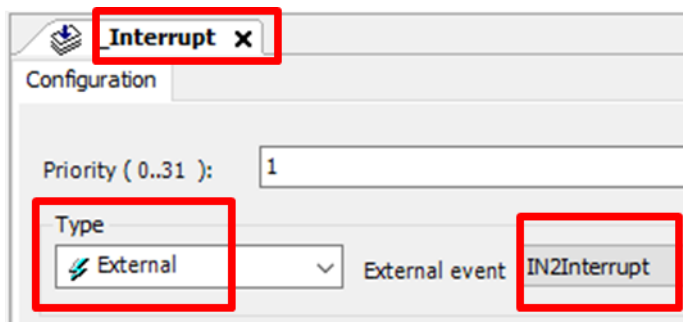
В начале необходимо создать Задачу (Task) обработки прерывания. Для этого встаньте в древе проекта на пункт **Task Configuration** и правой кнопкой мышки вызовите меню, в котором выберите пункт **Add Object – Task**:



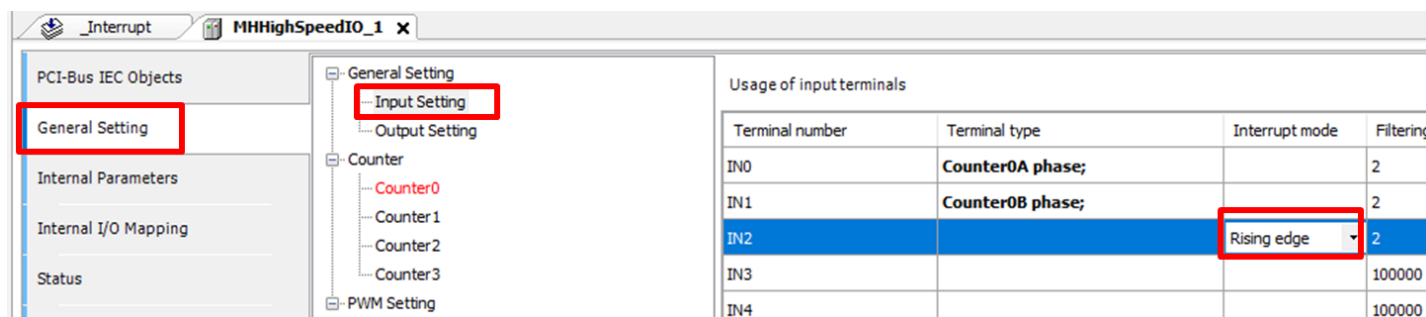
Дайте название Задаче, например **_Interrupt**. В древе проекта появится Задача **_Interrupt**:



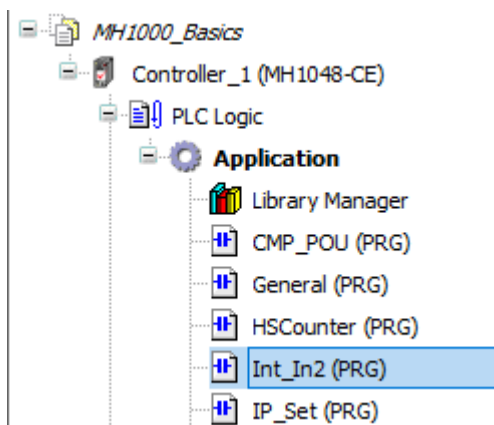
Щёлкните дважды левой кнопкой мышки на пункте **_Interrupt**, откроется вкладка с настройками параметров Задачи. Выберите тип задачи **External** и назначьте вход, например 02 (**IN2Interrupt**):



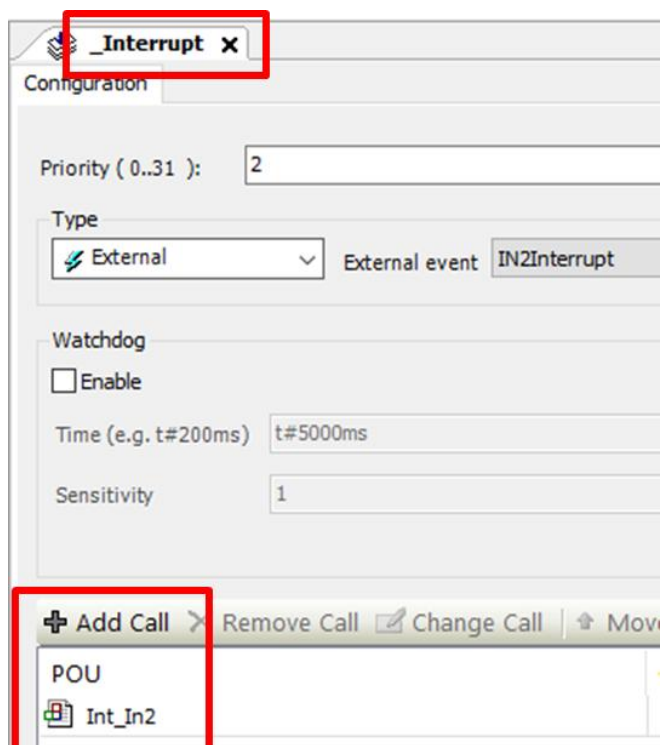
Щёлкните дважды на пункте **MHHighSpeedIO** и в открывшейся вкладке выберите раздел **General Configuration – Input Setting**. Напротив входа IN2 появится меню выбора фронта сигнала: Передний, задний или оба. На картинке ниже выбран передний фронт (**Rising edge**):



Далее создайте программный блок (POU), в котором необходимо поместить код, который должен быть исполнен по наступлению события, в нашем случае переднего фронта на входе 02. Дайте название, например **Int_In2**:

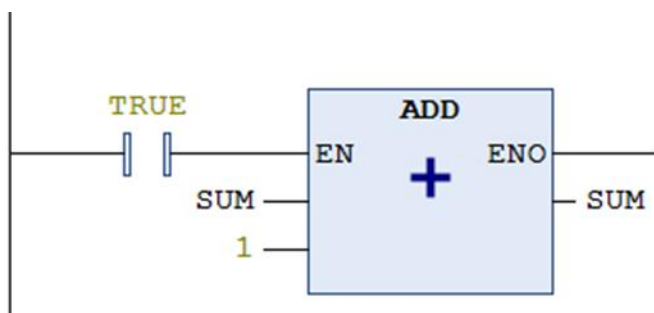


Созданный программный блок Int_In2 необходимо привязать к Задаче **_Interrupt**.

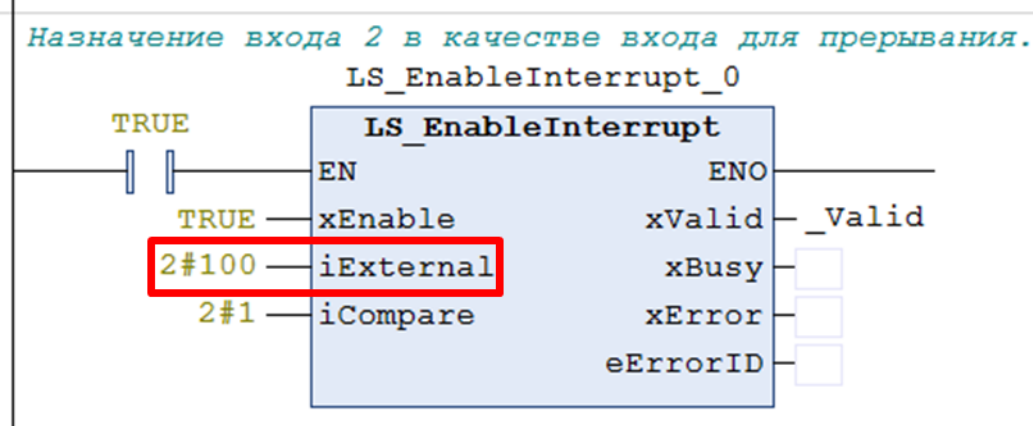


В теле POU **Int_In2** создайте одну строчку программы чисто в целях демонстрации работы прерывания:

Scope	Name	Address	Data type
VAR	SUM		INT



Далее необходимо зарегистрировать прерывание в команде **LS_EnableInterrupt**, которую необходимо поместить в циклической задаче (т.е. не в самом прерывании !!!):



На ножке **iExternal** в двоичном коде указываются входы ЦПУ, которые назначены для работы в режиме прерывания. В нашем примере это вход 02, поэтому бит 2 взводится в 1. Нумерация справа на лево.

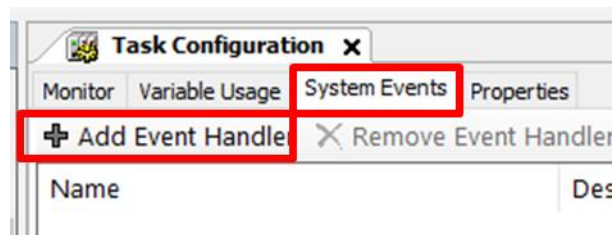
После загрузки в контроллер можно проверить работу вышеприведённой процедуры.

Если бы команда **ADD** стояла в обычной циклической Задаче, то она бы исполнялась в каждом скане и переменная **SUM** быстро бы увеличивалась. Но данный код привязан к Задаче по прерыванию, следовательно исполняться будет разово при наступлении соответствующего события – в нашем случае переднего фронта на входе 02.

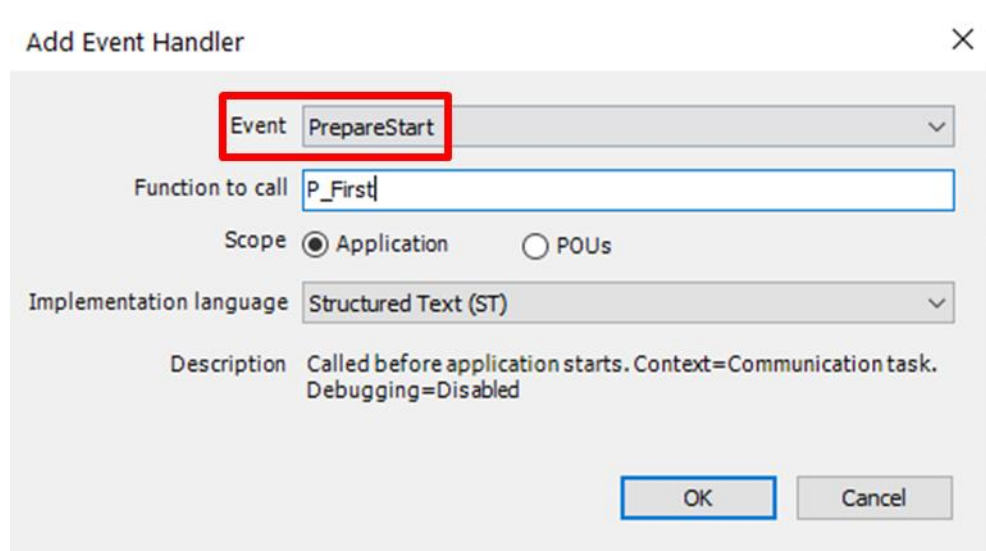
Импульс при переводе в состояние RUN

Практически во всех проектах возникает необходимость разового исполнения каких-либо действий при переводе контроллера в состояние Работа (RUN), например присвоить какие-либо константы и т.п. В среде CODESYS нет готового специального флага, который мог бы выполнить данную задачу. Поэтому ниже приводится процедура, которая позволяет реализовать данный функционал.

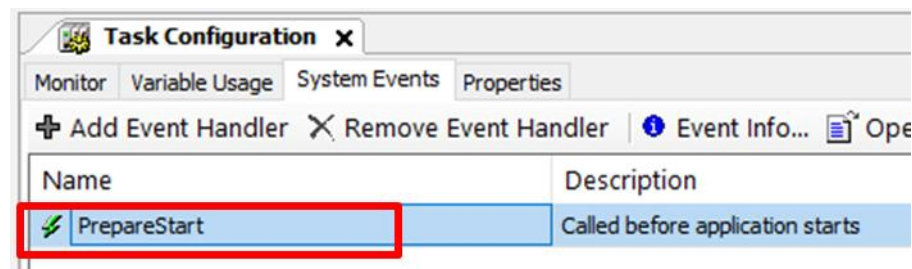
Для этого встаньте в древе проекта на пункт **Task Configuration** и двойным щелчком левой кнопки мышки откройте вкладку **Task Configuration** и выберите пункт **System Events** и нажмите кнопку **Add Event Handler**:



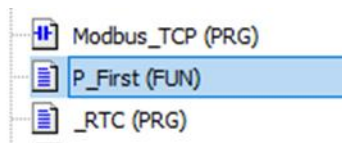
В открывшемся окне выберите тип события **Event** => **PrepareStart**, дайте имя, например **P_First** и нажмите **OK**.



В списке обработчиков появится пункт **PrepareStart**:



а в древе проекта появится функция **P_First**:



Далее откройте Таблицу Глобальных Переменных (GVL) и создайте глобальную переменную **P_First**:

	Scope	Name	Address	Data type
1	VAR_GLOBAL	P_First		BOOL

Откройте двойным щелчком функцию **P_First** и напишите одну строчку кода:

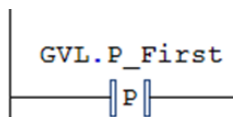
```
1 GVL.P_First:=TRUE;
```

В таблице переменных система автоматически уже создала переменную:

	Scope	Name	Address	Data type
1	VAR_IN_OUT	EventPrm		CmpApp.EVTPARAM_CmpApp

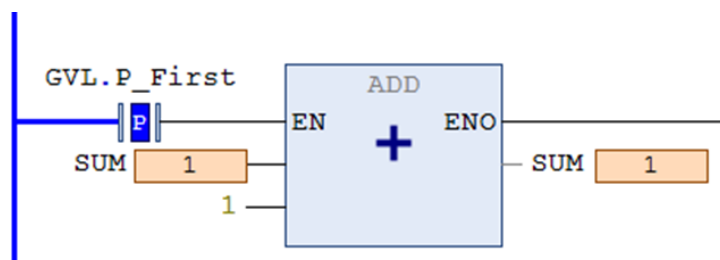
никаких других переменных в функции-обработчике создавать не надо!

На этом процедура оформления контакта, замкнутого один раз при переводе контроллера в работу, окончена. В качестве контакта выступает глобальная переменная **P_First**, которую необходимо оформить как контакт по переднему фронту:



Для проверки работоспособности данной процедуры можно в любой циклической задаче написать простейший код типа:

	Scope	Name	Address	Data type
1	VAR	SUM		INT



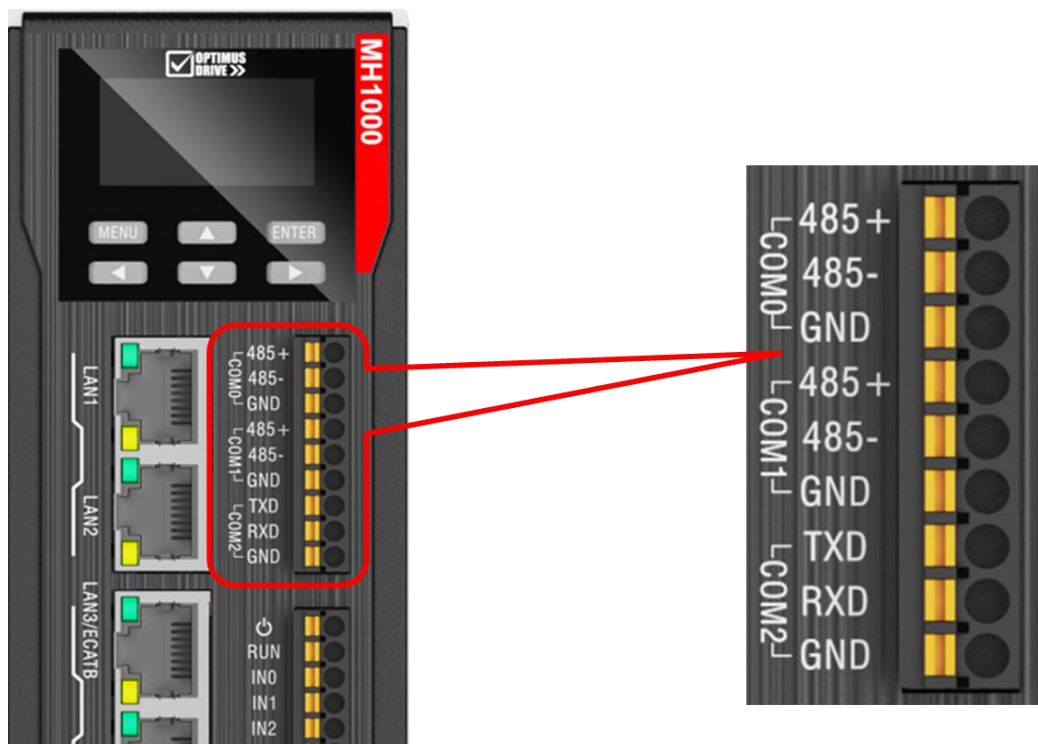
Переменная SUM будет равна 1 на всё время работы контроллера.

Последовательная связь по протоколу Modbus RTU Master

Контроллеры серии MH1000 имеют на борту 1 порт RS232 и 2 порта RS485. Все 3 порта могут работать как в режиме Мастера, так и Водомого. В данной главе рассмотрена работа в режиме Мастера по протоколу Modbus RTU.

Физически порты имеют на контроллере следующее расположение:

COM0 -RS485, COM1 – RS485, COM2 – RS232



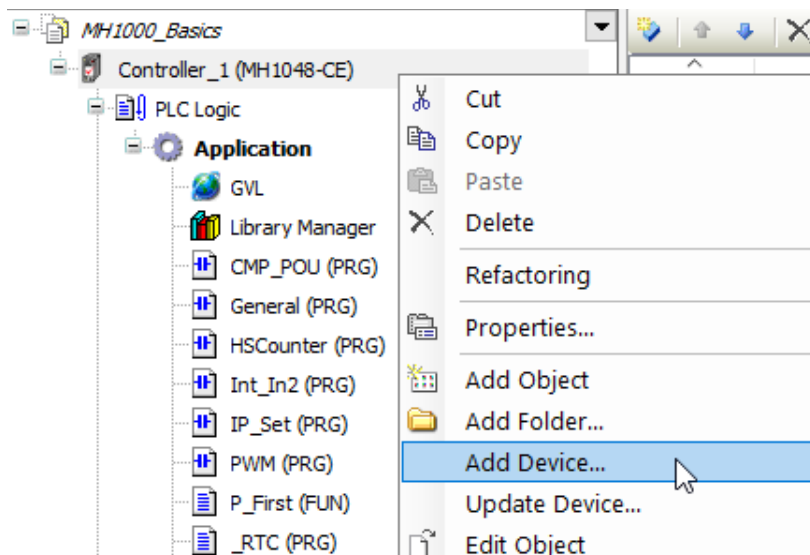
По RS232 возможно соединение только точка-точка. Максимальная длина кабеля на скорости 9600 максимум 15 метров. На скорости 115200 – 3 метра.

По RS485 возможно подключение до 31 Водомого. 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.

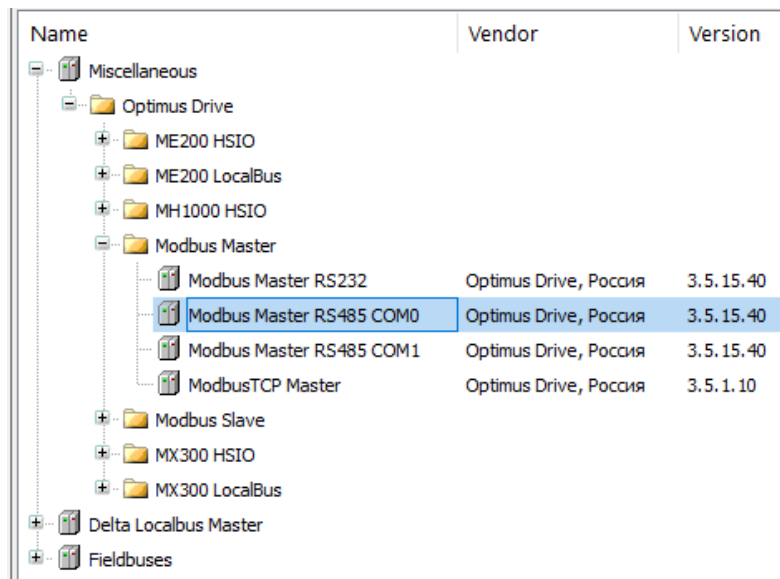


Внимание! ЦПУ содержит встроенные резисторы 120 Ом для портов RS485 COM0 и COM1. Отключить их нельзя. Данный факт необходимо учитывать при проектировании сети RS485 (контроллер должен быть в начале (конце) линии).

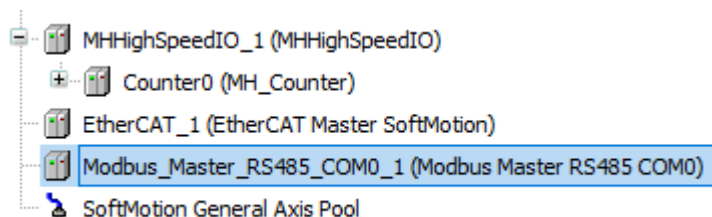
Для организации связи по Modbus RTU в проект необходимо добавить адаптер соответствующего порта. Щёлкните правой кнопкой мышки на вкладке **Device** и в открывшемся меню выберите пункт **Add Device**:



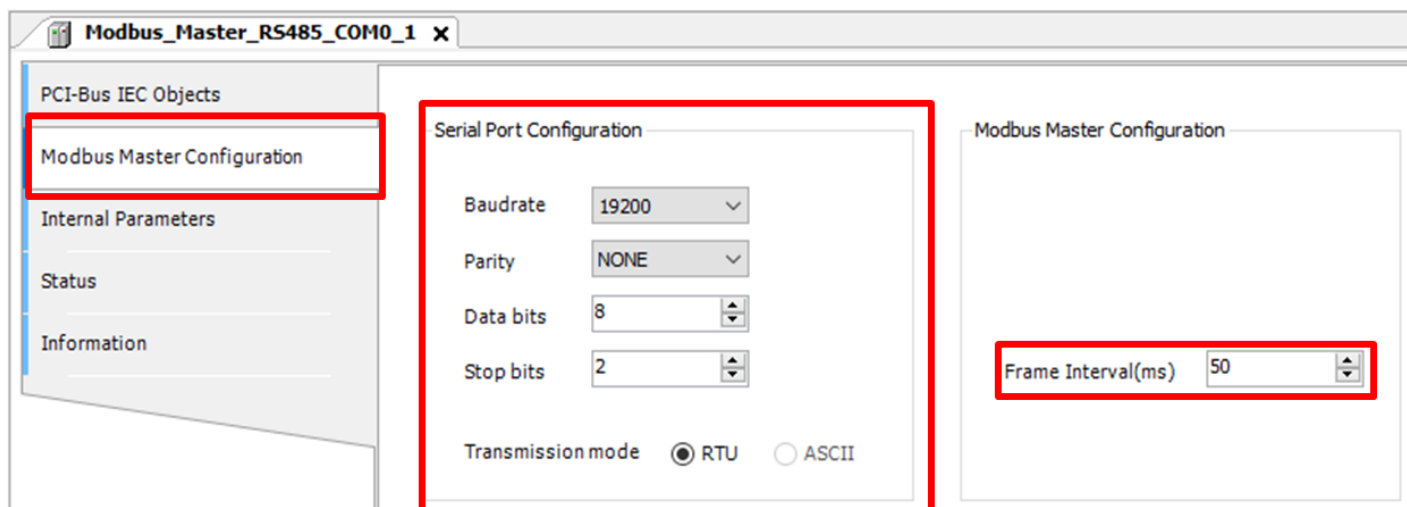
В открывшемся окне выберите папку **Miscellaneous – Optimus Drive – Modbus Master – Modbus Master RS485 COM0**:



В древе проекта появится пункт **Modbus_Master_RS485_COM0_1**:

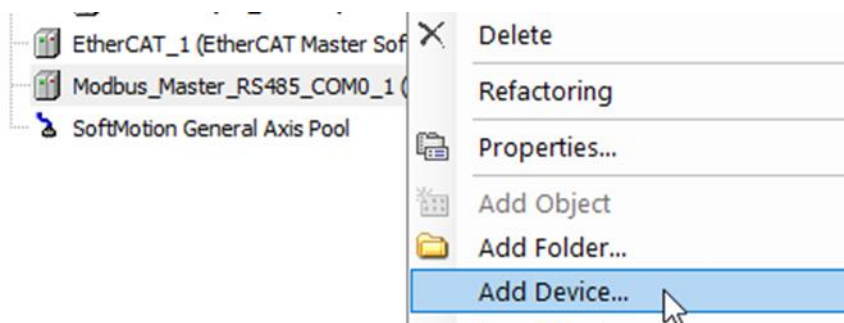


Щёлкните по этому пункту дважды левой кнопкой мышки и в открывшейся вкладке сделайте настройки связи в разделе **Modbus Master Configuration** для порта RS485_COM0:

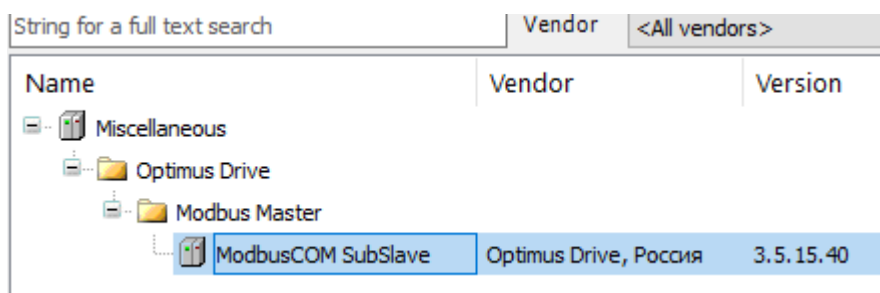


Поддерживается только режим Modbus RTU. В нашем примере выставлена скорость 19200, протокол 8, N, 2. Пункт справа **Frame Interval (ms)** определяет темп отправки пакетов. В нашем примере стоит 50 мс.

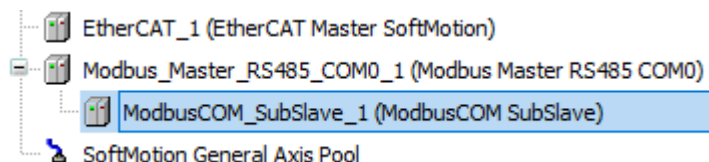
Далее щёлкните на пункте Modbus_Master_RS485_COM0_1 правой кнопкой мышки и в открывшемся окне выберите пункт **Add Device** и добавьте Ведомое устройство **Modbus RTU**:



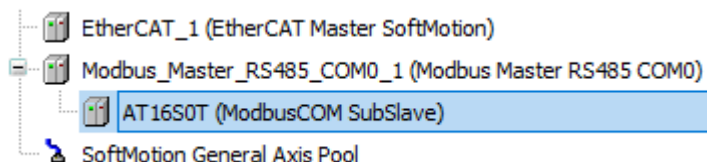
В открывшемся окне выберите пункт **ModbusCOM_SubSlave**:



В древе проекта появится пункт **ModbusCOM_SubSlave_1**:

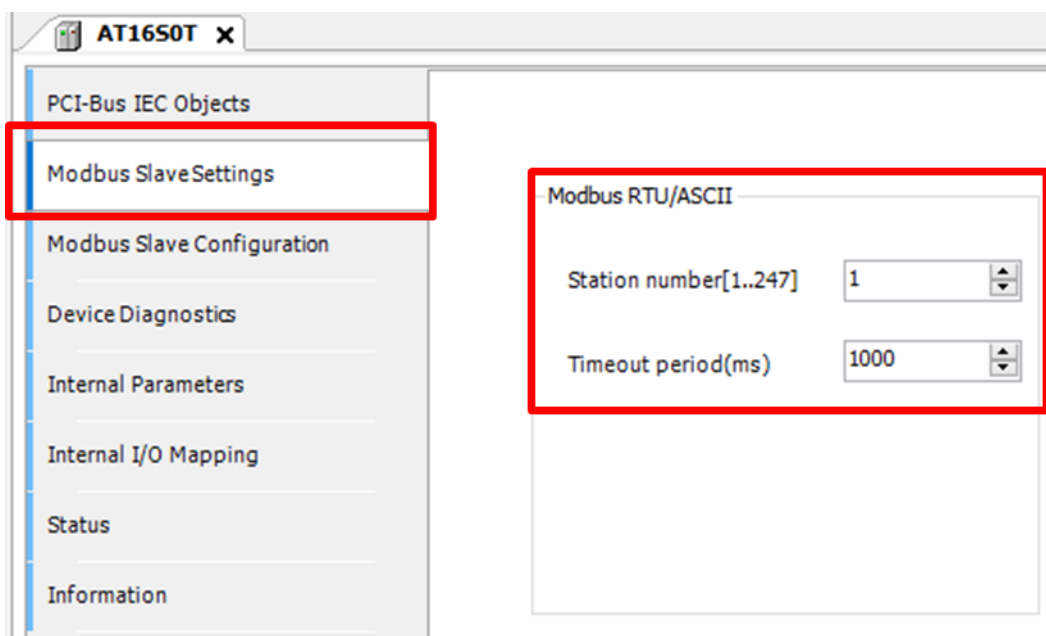


Название Ведомого можно поменять. Например заменить на контроллер AT16S0T:

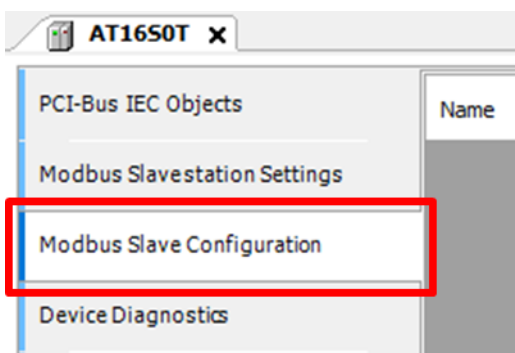


Щёлкните дважды на Ведомом, откроется вкладка для создания запросов.

В пункте **Modbus Slave Station Settings** необходимо задать сетевой адрес Ведомого и таймаут связи:



Далее в пункте **Modbus Slave Configuration** необходимо задать регистры Ведомого и их количество для чтения/записи Мастером. Всего может быть создано 31 соединение с Ведомым.

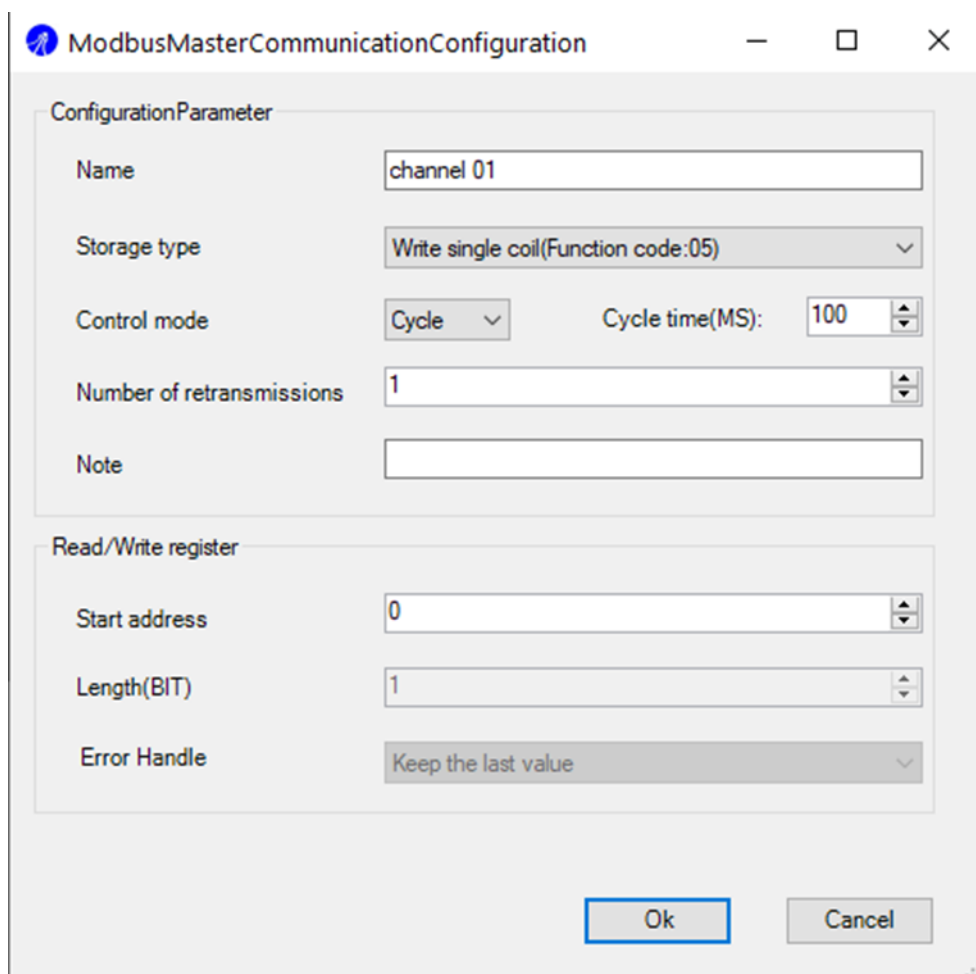


Для создания поля обмена данным с Ведомым необходимо нажать кнопку **Add**:

Name	Configuration number	R/W types	Control mode	shifting	Length	Error handle	Number of retransmissions	Note

Add Delete Edit Import Export

Появится поле обмена данными с Ведомым, в котором необходимо определить регистры Ведомого, их количество и регистры Мастера для чтения/записи данных:



ModbusMasterCommunicationConfiguration

ConfigurationParameter

Name: channel 01

Storage type: Write single coil(Function code:05)

Control mode: Cycle Cycle time(MS): 100

Number of retransmissions: 1

Note:

Read/Write register

Start address: 0

Length(BIT): 1

Error Handle: Keep the last value

Ok Cancel

Открывшееся окно содержит следующие настройки:

Name – Имя запроса (латинские буквы и цифры)

Storage Type – Код функции Modbus

Control Mode – Метод отправки запросов. **Cyclic** – циклически в автоматическом режиме, **Trigger** – запрос отправляется по переднему фронту назначенного регистра. Регистр можно посмотреть во вкладке **Internal I/O Mapping**. При выборе режима Trigger появится переменная **Trigger control bit**.

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

Number of retransmissions – Количество повторных запросов

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

Read/Write Register – блок, в котором оформляется чтение/запись данных из Ведомого

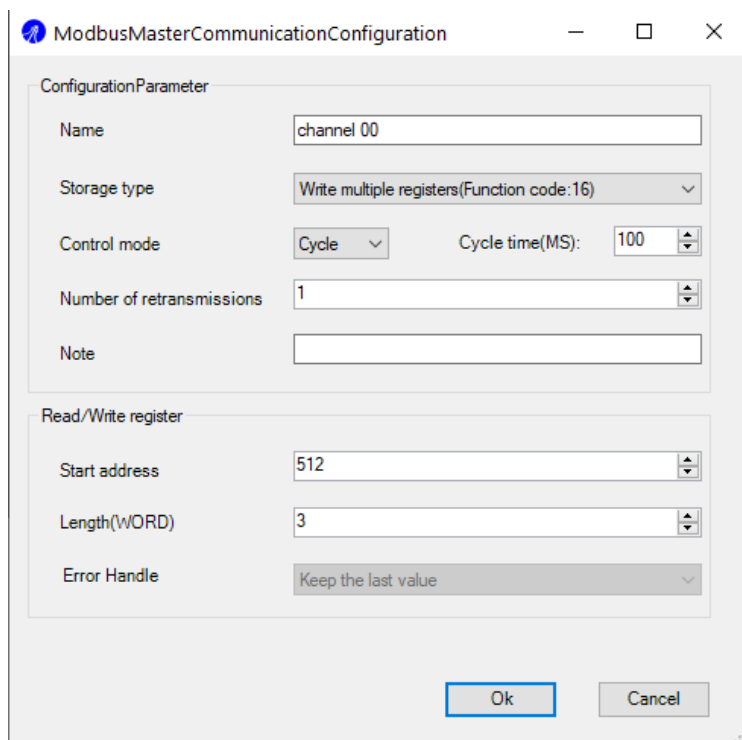
Start Address – Адрес регистра Ведомого, задаётся в HEX записью вида 0x0000

Length – Длина данных для чтения/записи

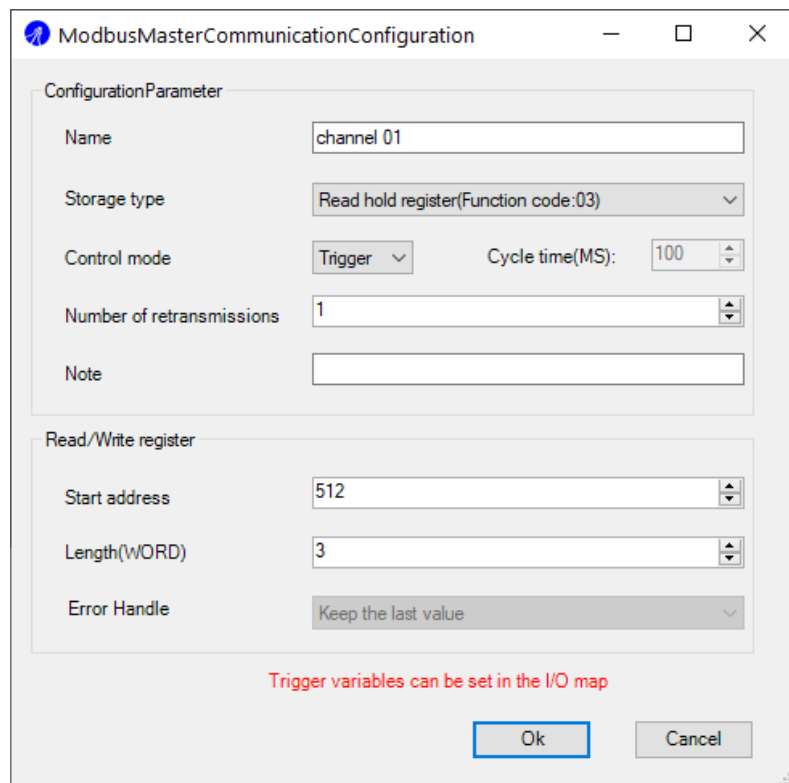
Error Handling – Фиксировано на сохранять состояние регистров при потере связи

Ниже для примера оформлено два запроса Channel_0 и Channel_1 для записи и чтения регистров V0, V1 и V2 контроллера Optimus Drive AT16S0T-RU, который выступает в качестве Ведомого устройства

Запись. Имя – Channel 00, Команда 16 (групповая запись регистров), метод опроса – Cycle циклический с тактом в 100 мс, Начальный регистр для чтения задаётся в десятичном виде без смещения – 512 (0x0200), количество последовательно записываемых регистров 3 штуки.



Чтение. Имя – Channel 01, Команда 3 (групповое чтение регистров), метод опроса – Trigger, из программы при помощи переменной. Начальный регистр для чтения задаётся в десятичном виде без смещения – 512 (0x0200) соответствует регистру V0 в контроллере AT16S0T, количество последовательно считываемых регистров 3 штуки.



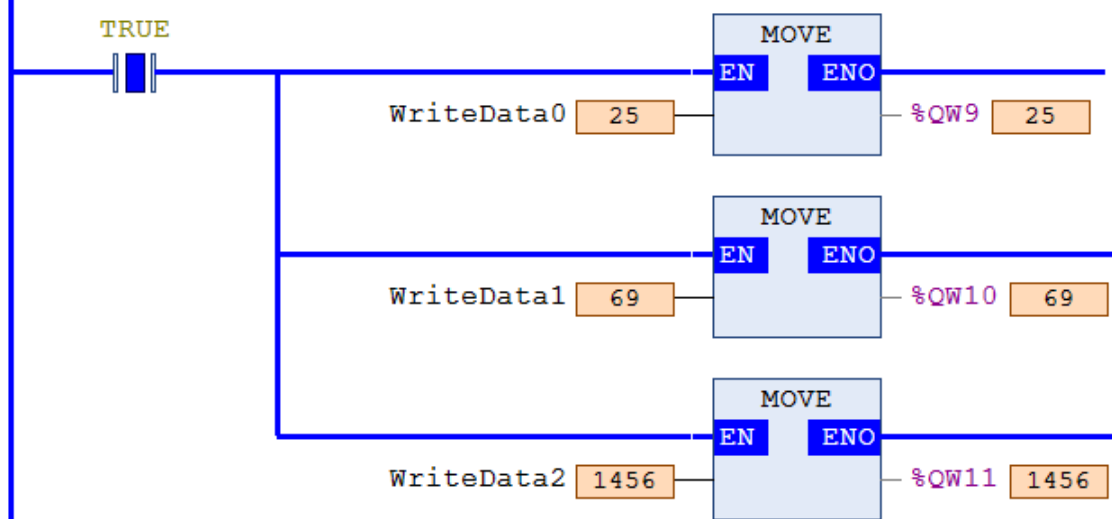
После создания каналов опроса система автоматически выделит под них регистры в соответствии с заявленным количеством регистров в Ведомом устройстве. В нашем примере будет 3 регистра на запись и 3 на чтение. Посмотреть можно в пункте **Internal I/O Mapping**:

Channel	Address	Type	Unit	Description
ErrorCode	%IW36	WORD		ModbusRTU ErrorCode
channel 00	%QW9	ARRAY [0..2] OF WORD		Write multiple registers
channel 00[0]	%QW9	WORD		Write #0200
channel 00[1]	%QW10	WORD		Write #0201
channel 00[2]	%QW11	WORD		Write #0202
channel 00	%IX74.0	BOOL		Processing complete state
channel 01	%IW38	ARRAY [0..2] OF WORD		Read hold register
channel 01[0]	%IW38	WORD		Read #0200
channel 01[1]	%IW39	WORD		Read #0201
channel 01[2]	%IW40	WORD		Read #0202
channel 01	%IX82.0	BOOL		Processing complete state
channel 01	%QX24.0	BOOL		Trigger control bit

Приведённые выше регистры можно напрямую использовать в программе для отправки/приёма данных от ведомого устройства. Сам обмен будет осуществляться автоматически. Также, можно создать свои переменные поверх выделенных системой регистров.

Запись данных в Ведомое устройство (Write). Запросы отправляются циклически без дополнительных условий.

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

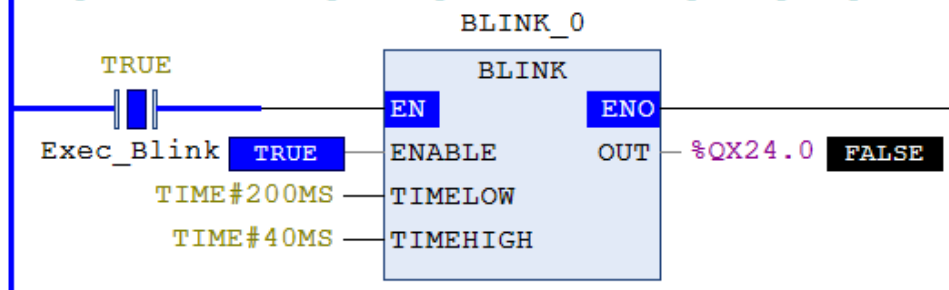


Приём данных от Ведомого устройства (Read). Запросы отправляются по триггеру:

Channel V2	Address	Value	Processing complete state
channel 01	%QX24.0	BOOL	Trigger control bit

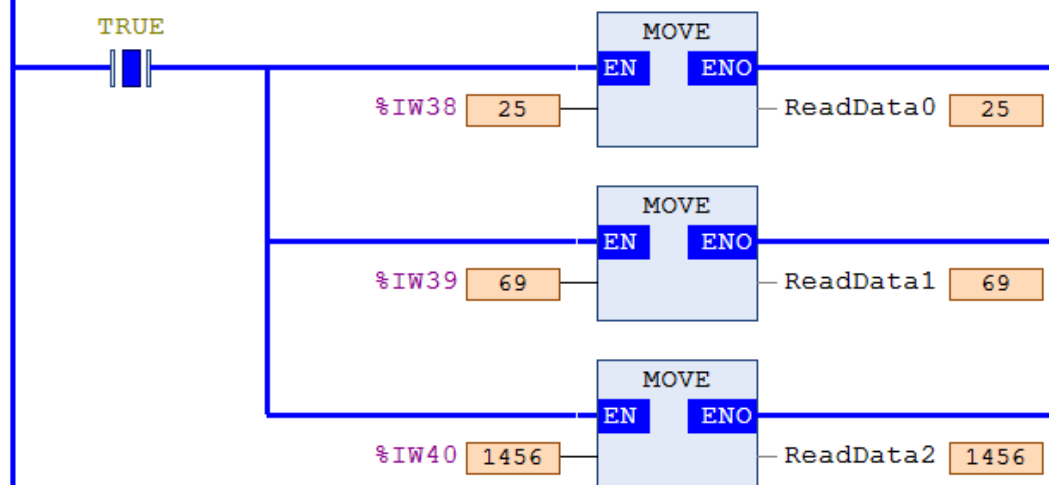
Триггер можно организовать при помощи команды Blink:

Запрос на чтение регистров V0-V2 контроллера Optimus Drive AT16S0T-RU.

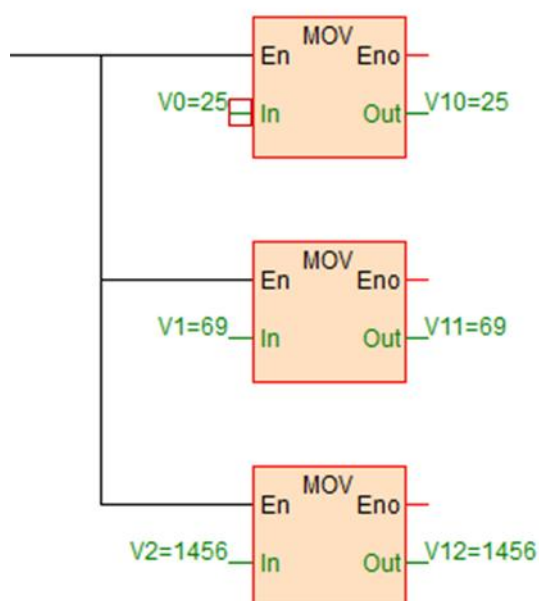


Принятые данные от Ведомого:

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



Мониторинг программы контроллера AT16S0T-RU:

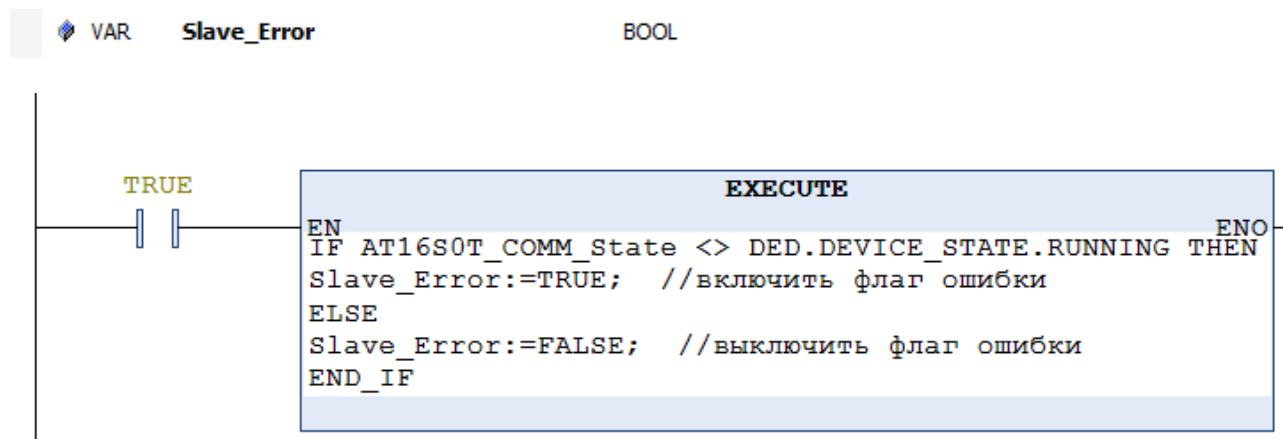


Ошибки связи можно проверять через свойство узла **GetDeviceState** и переменную типа **DED.DEVICE_STATE**

В нашем примере это переменная **AT16S0T_COMM_State**:

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

В программе состояние связи с Ведомым удобно проверять с помощью кода в блоке **Execute**, в котором используется переменная **Slave_Error**:



Текст программы из блока Execute:

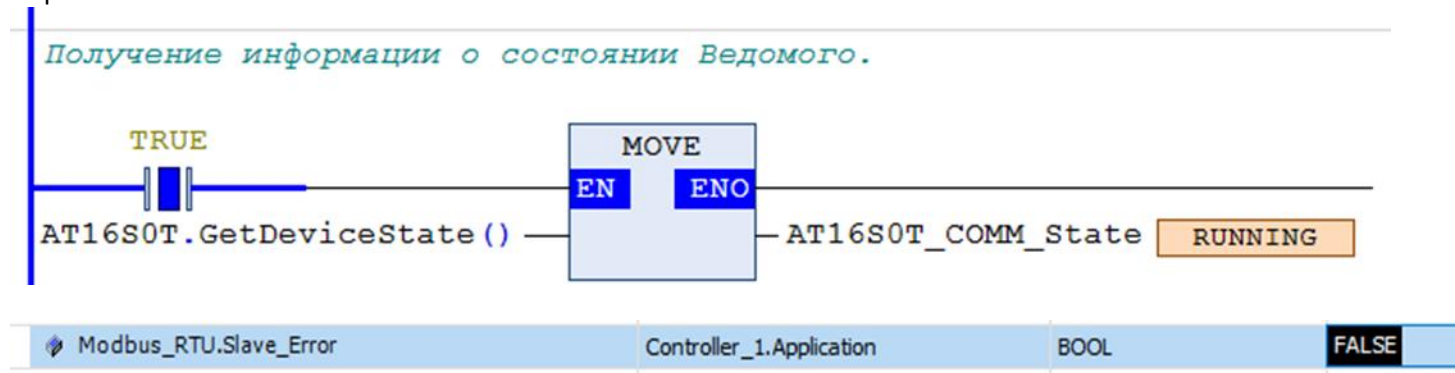
```

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

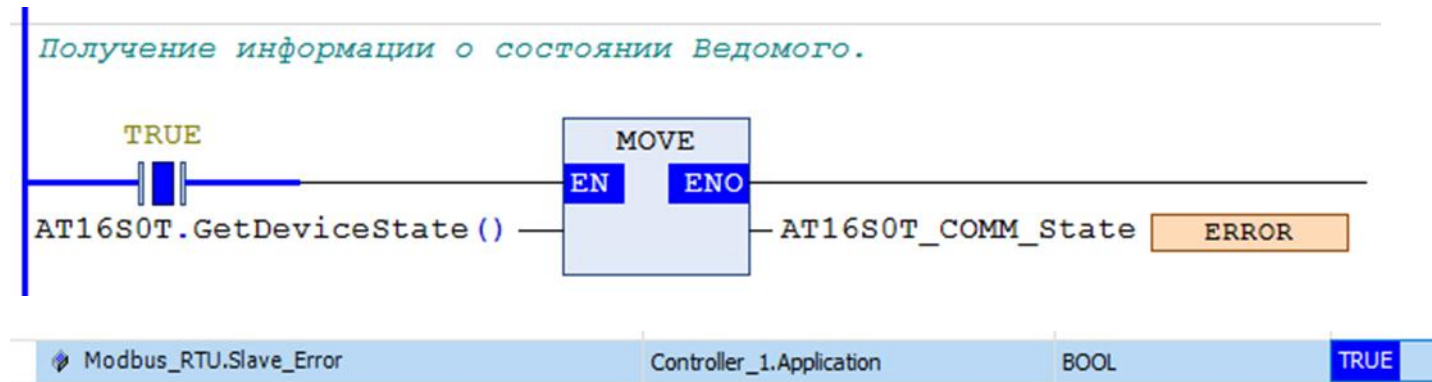
```

Мониторинг состояния Ведомого в программе:

Нормальное состояние:



Ошибка связи:

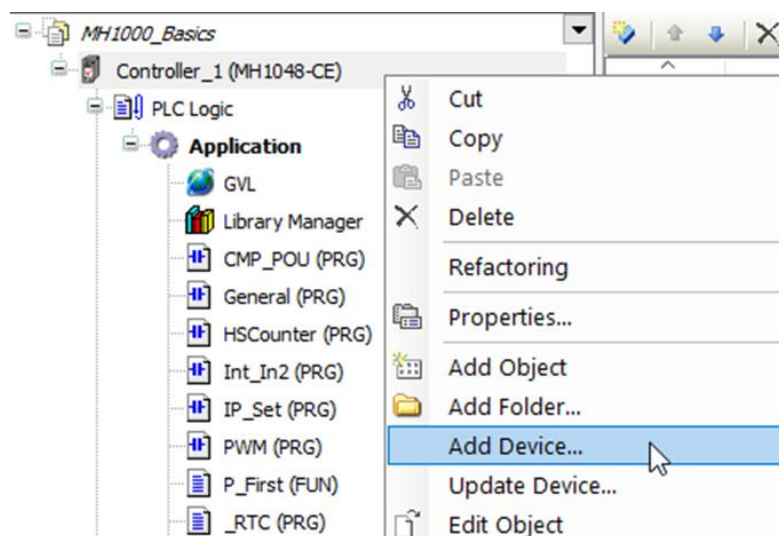


Последовательная связь по протоколу Modbus RTU Slave

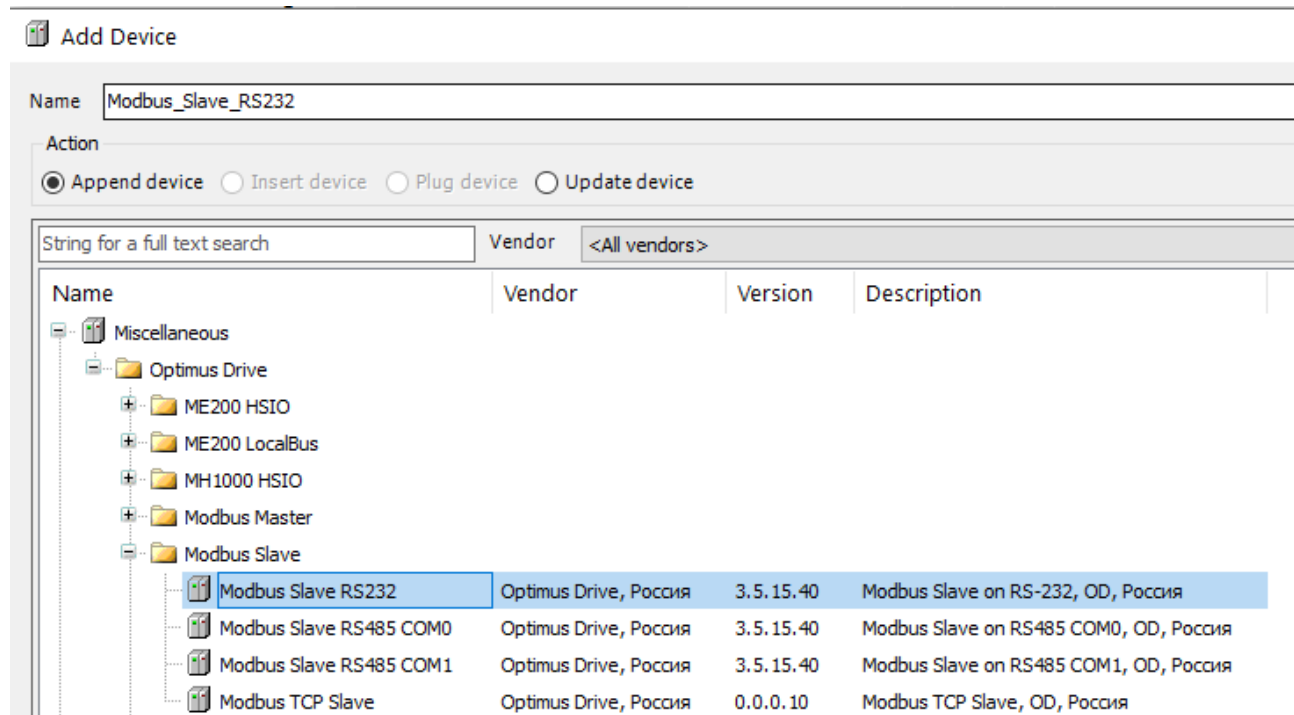
Контроллеры серии MH1000 имеют на борту 1 порт RS232 и 2 порта RS485. Все 3 порта могут работать как в режиме Мастера, так и Водомого. В данной главе рассмотрена работа в режиме Водомого по протоколу Modbus RTU.

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**:

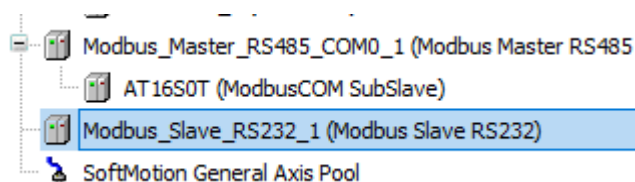


В открывшемся окне выберите пункт **Miscellaneous** и адаптер нужного порта, например RS232:



Нажмите **ОК**.

В древе проекта появится пункт **Modbus_Slave_RS232**:



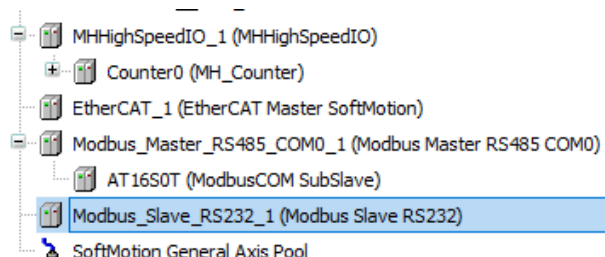
После добавления адаптера в проект и его загрузки, контроллер откроет доступ к своей памяти по протоколу Modbus RTU через соответствующий порт. Адресация будет одинаковая для всех портов. Согласно спецификации протокола Modbus, он имеет возможность читать булевы и словные регистры без обозначения типов 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

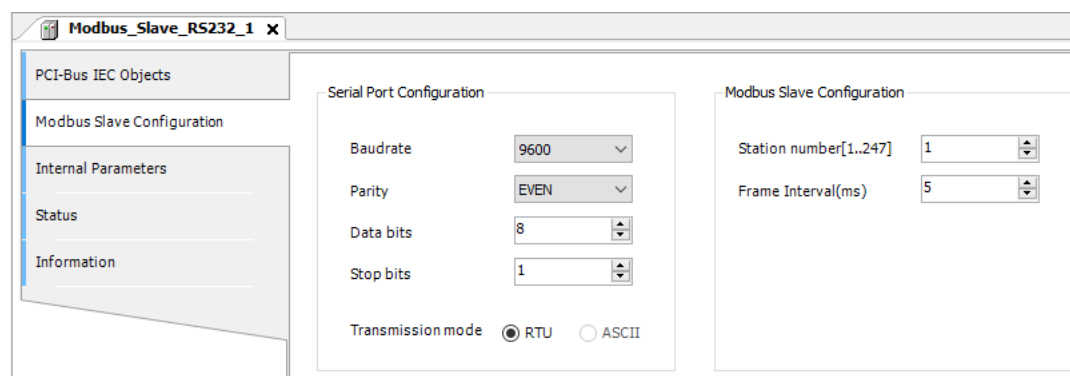
Регистры типа %MW0~%MW65535 являются стандартными словами 16 бит. Адресация к битам в словах не поддерживается. Поэтому в программе биты типа %MX0.0... можно использовать, но их состояние нужно передавать как словный регистр, а потом на стороне Мастера разбирать по битам.

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 **Modbus_Slave_RS232**:



В открывшейся вкладке выберите пункт **Modbus Slave Configuration** и сделайте нужные настройки скорости и формата передачи данных:



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 MH1000 не поддерживается.

Контроллеры серии MH1000 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 скорость и протокол:

Communication Setting

Communication Type: RS485-2

Baud Rate: 9600

Data Bit: 8

Stop Bit: 1

Parity Bit: None

и во вкладке Advance убрать смещение адреса:

Protocol Timeout2: 0

Max Bit Registers: 64

Time Interval: 30

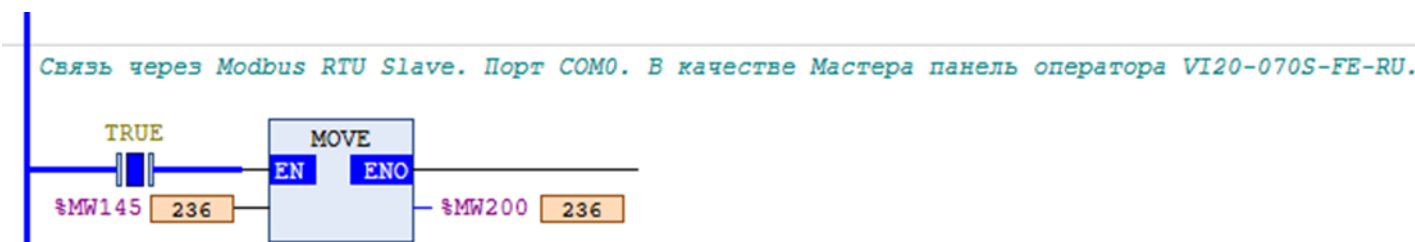
Base Address: 0

32-bit Integer: 4321

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

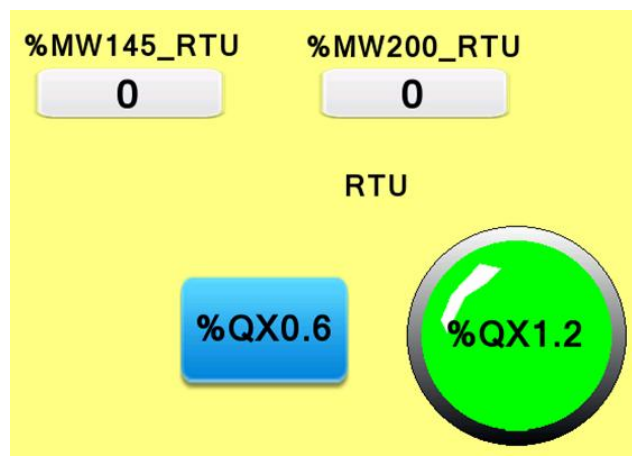


Регистр %QX0.6 - контакт, регистр %QX1.2 - выходная катушка.



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:

☐ Use Address Tag

Device:

Station No: ☐ Index

Address Type:

Address:

Format(Range):DDDDD(0~65534)

а к регистру %MW200 так

Device:

Station No: ☐ Index

Address Type:

Address:

Таким образом, нужно просто указать номер регистра %MW****. Обращение к булеву регистру %QX0.6 выглядит так:

Device:

Station No: ☐ Index

☐ Bit-index within a Byte Register

Address Type:

Address:

Format(Range) DDDDD(0~65534)

а к регистру %QX1.2 так:

Device:

Station No: ☐ Index

☐ Bit-index within a Byte Register

Address Type:

Address:

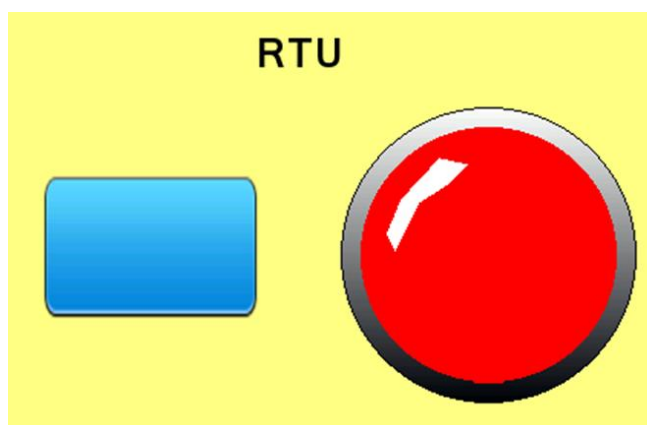
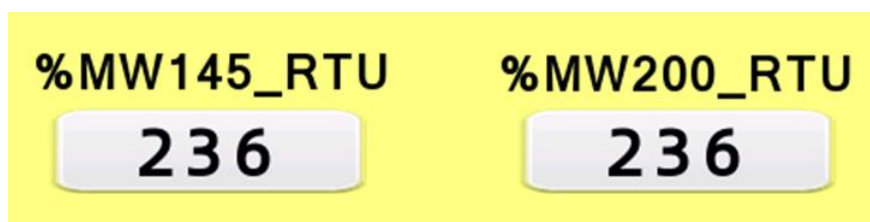
Format(Range):DDDDD(0~65534)

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 МН1048 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).

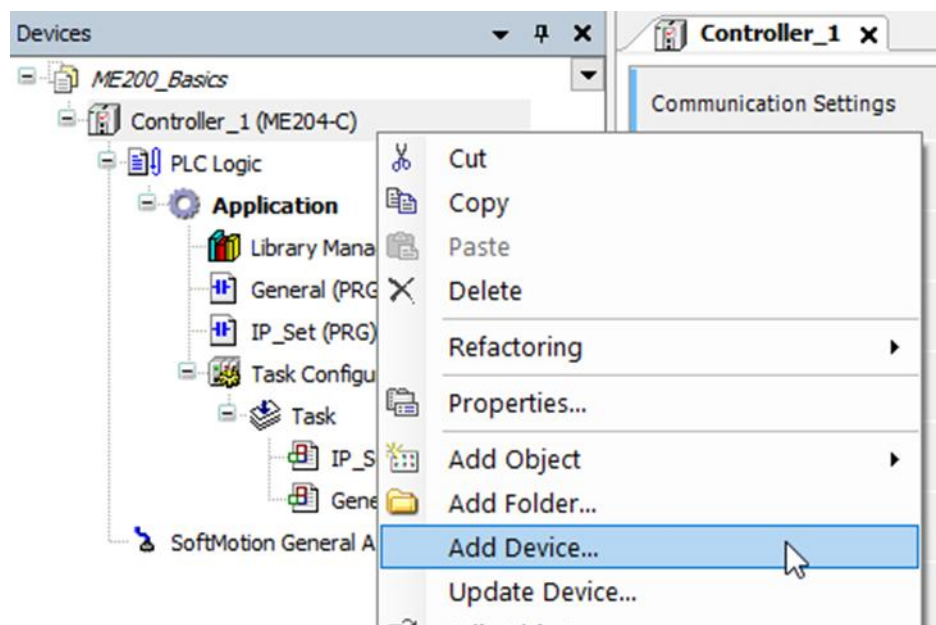
Для использования Modbus TCP Client (Master) к проекту должны быть подключены следующие библиотеки: **IoDrvEthernet**, **IoDrvModbusTCP** и **DED**.

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**:



Выберите пункт **Miscellaneous – Optimus Drive – Modbus Master – Modbus TCP Master**:

Add Device

Name:

Action:

☒ Append device ☐ Insert device ☐ Plug device ☐ Update device

String for a full text search: Vendor:

Name	Vendor	Version
Miscellaneous		
Optimus Drive		
ME200 HSIO		
ME200 LocalBus		
MH1000 HSIO		
Modbus Master		
Modbus Master RS232	Optimus Drive, Россия	3.5.15.40
Modbus Master RS485 COM0	Optimus Drive, Россия	3.5.15.40
Modbus Master RS485 COM1	Optimus Drive, Россия	3.5.15.40
ModbusTCP Master	Optimus Drive, Россия	3.5.1.10
Modbus Slave		

Нажмите кнопку **Add Device**:

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

Modbus_Slave_RS232_1 (Modbus Slave RS232)

ModbusTCP_Master_1 (ModbusTCP Master)

SoftMotion General Axis Pool

Щёлкните дважды левой кнопкой мышки на пункте **Modbus TCP Master**, откроется вкладка настроек параметров Мастера:

ModbusTCP_Master_1

PCI-Bus IEC Objects

ModbusTcp MasterSettings

Internal Parameters

Status

Information

ModbusTcp Config

Response timeout(ms):

Socket interval(ms):

☒ Automatic reconnection

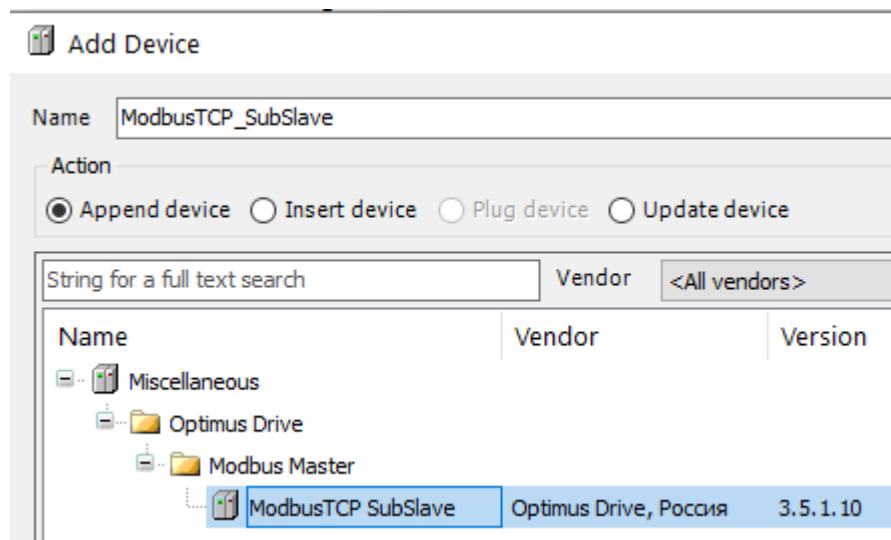
В разделе **Modbus TCP Master Settings** необходимо задать:

Таймаут связи мс (Response Time (ms))

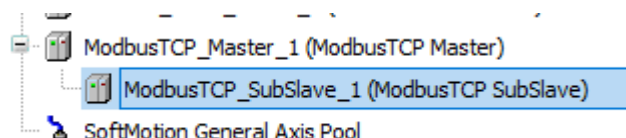
Задержка отправки пакетов мс (Socket interval (ms))

Автоматическое восстановление связи (Automatic reconnection) поставить флажок

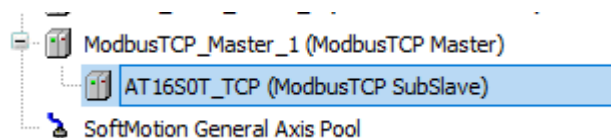
Контроллер готов к работе по протоколу Modbus TCP Client (Master) и осталось добавить линки, т.е. конкретные ведомые устройства, которые контроллер будет опрашивать. Для этого встаньте мышкой на пункт **Modbus TCP Master** и щёлкните правой кнопкой мышки и в отрывшемся меню выберите пункт **Add Device**.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



У Вас в древе проекта появится пункт с Ведомым устройством,

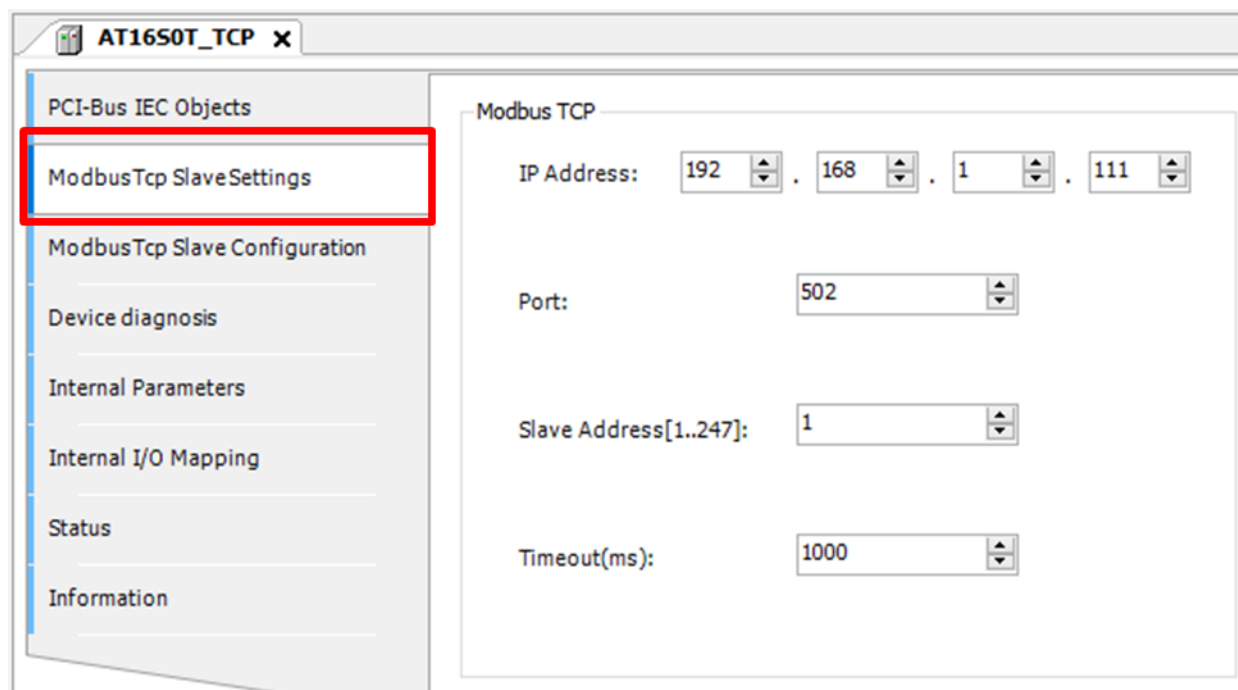


которому можно сразу переименовать на нужное название Ведомого. В нашем примере в качестве Ведомого будет использоваться контроллер Optimus Drive AT16S0T-RU

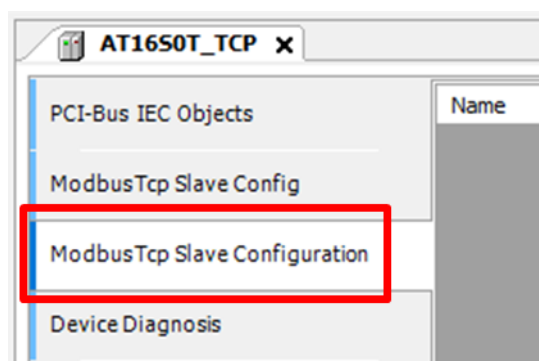


Имя узла - AT16S0T_TCP

Щёлкните дважды левой кнопкой мышки на Ведомом, откроется вкладка настройки обмена с Ведомым. В разделе **Modbus TCP Slave Config** установите IP адрес Ведомого (в нашем примере 192.168.1.111), Modbus адрес Ведомого (Slave Address), порт (Port), Таймаут (Timeout):

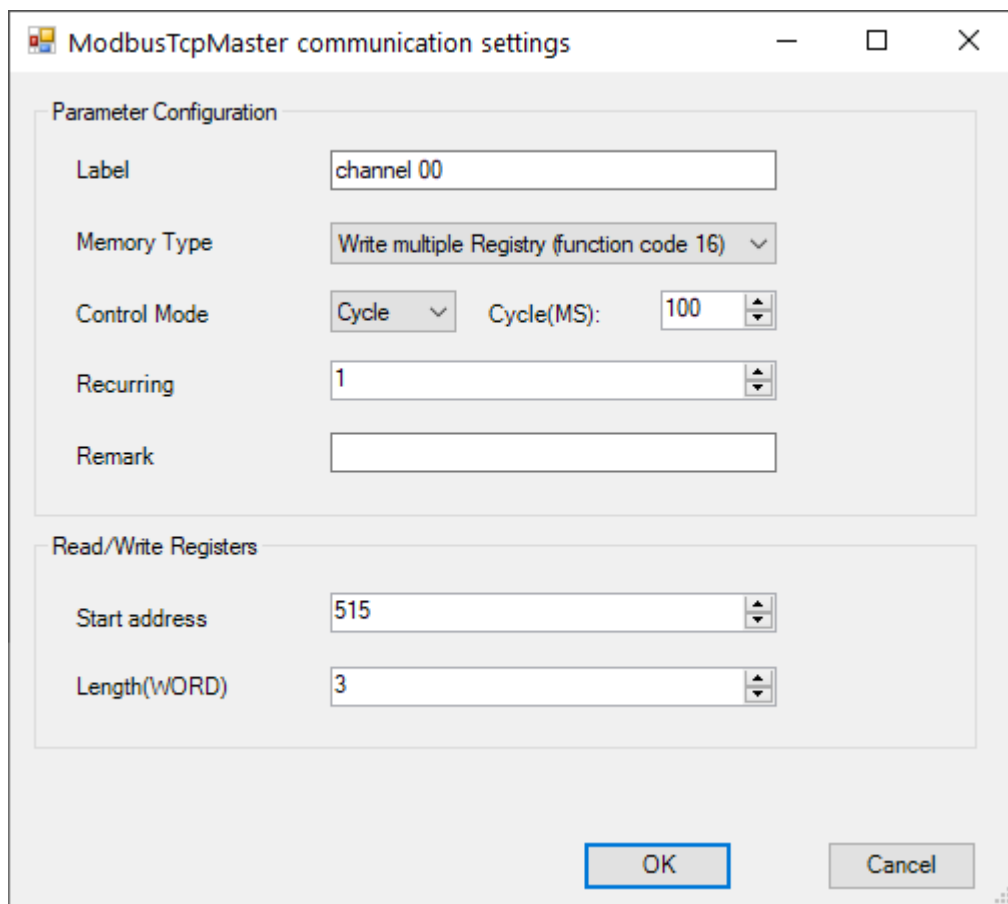


Далее во вкладке **Modbus Tcp Slave Configuration** необходимо создать нужное количество запросов:



Для оформления запроса нажмите кнопку **Add**. Откроется окно создания запроса:

Открывшееся окно содержит следующие настройки:



Label – Имя запроса (латинские буквы и цифры)

Memory Type – Код функции Modbus

Control Mode – Метод отправки запросов. **Cyclic** – циклически в автоматическом режиме, **Trigger**– запрос отправляется по переднему фронту назначенного регистра. Регистр можно посмотреть во вкладке **Internal I/O Mapping**. При выборе режима Trigger появится переменная **Trigger control bit**.

Cycle(MS) – Время цикла опроса в миллисекундах. Для метода **Cyclic**

Recurring – Количество повторных запросов

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

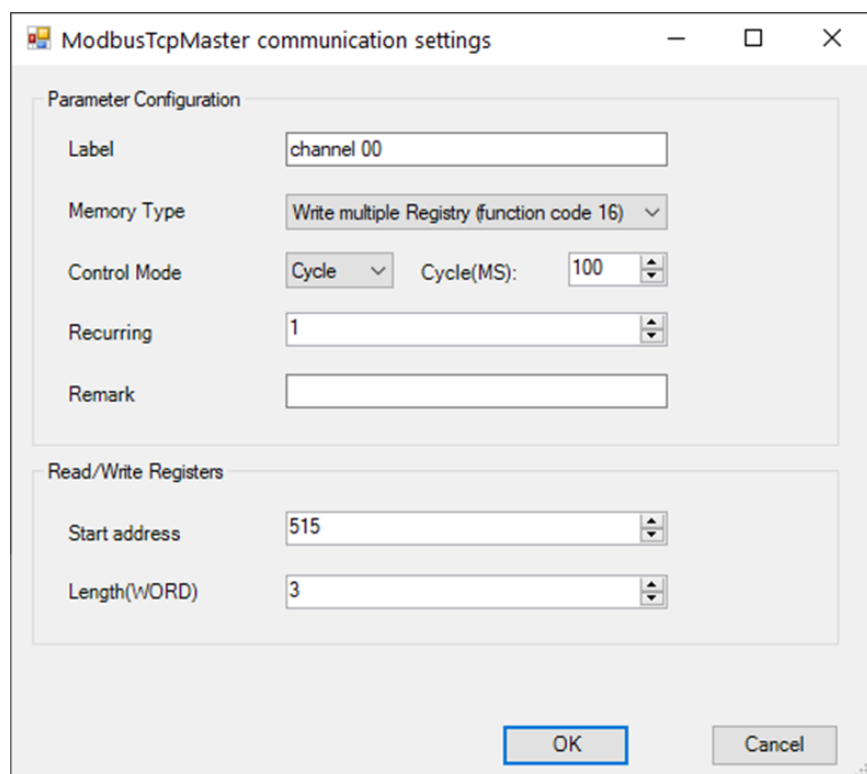
Read/Write Register – блок, в котором оформляется чтение/запись данных из Ведомого

Start Address – Адрес регистра Ведомого, задаётся в HEX записью вида 0x0000

Length – Длина данных для чтения/записи

Ниже для примера оформлено два запроса Channel 00 и Channel 01 для записи и чтения регистров V3-V5 контроллера Optimus Drive AT16S0T-RU

Запись (адрес регистра записывается в десятичном виде без смещения, 515 это регистр V3 (0x0203)).
Имя – Channel 00, Команда 16 (групповая запись регистров), метод опроса – циклический с тактом в 100 мс,
Начальный регистр для записи 515 (0x0203), количество последовательно записываемых регистров 3 штуки.



ModbusTcpMaster communication settings

Parameter Configuration

Label: channel 00

Memory Type: Write multiple Registry (function code 16)

Control Mode: Cycle Cycle(MS): 100

Recurring: 1

Remark:

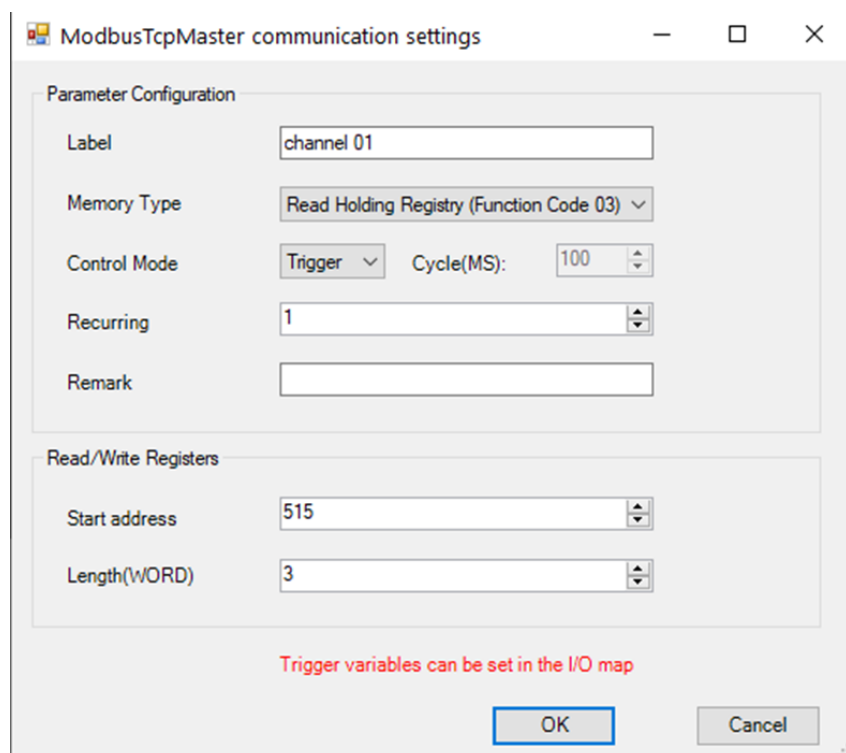
Read/Write Registers

Start address: 515

Length(WORD): 3

OK Cancel

Чтение. Имя – Channel 01, Команда 3 (групповое чтение регистров), метод опроса – Trigger, из программы битом. Начальный регистр для чтения 515 (0x0203), количество последовательно считываемых регистров 3 штуки.



ModbusTcpMaster communication settings

Parameter Configuration

Label: channel 01

Memory Type: Read Holding Registry (Function Code 03)

Control Mode: Trigger Cycle(MS): 100

Recurring: 1

Remark:

Read/Write Registers

Start address: 515

Length(WORD): 3

Trigger variables can be set in the I/O map

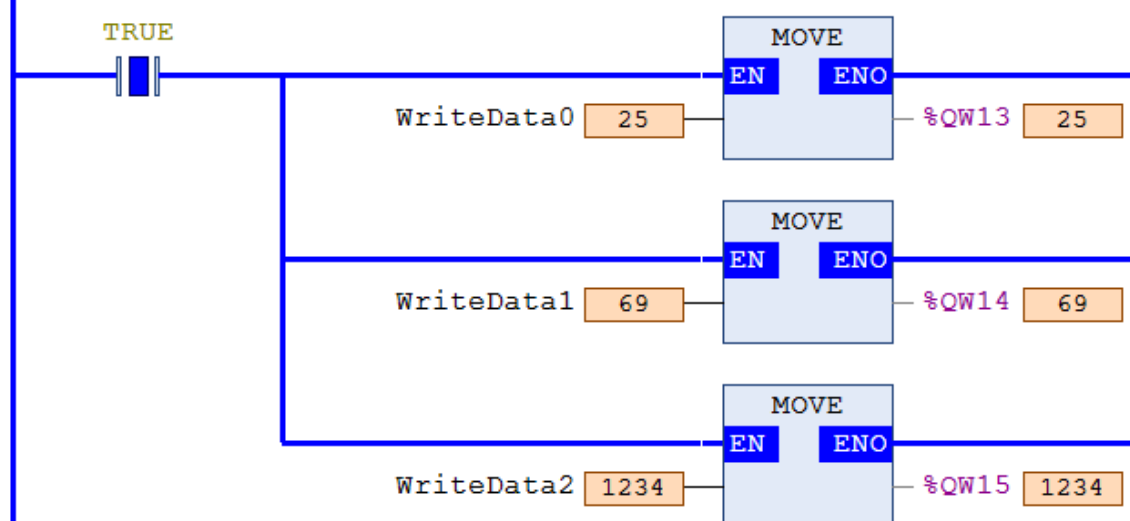
OK Cancel

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 **Internal I/O Mapping**:

Variable	Mapping	Channel	Address	Type	Unit	Description
		ErrorCode	%IW42	WORD		ModbusTCP ErrorCode
		channel 00	%QW13	ARRAY [0..2] OF WORD		Write multiple Registry (function code 16)
		channel 00[0]	%QW13	WORD		Write #0515)
		channel 00[1]	%QW14	WORD		Write #0516)
		channel 00[2]	%QW15	WORD		Write #0517)
		channel 00	%IX86.0	BOOL		Processing complete state
		channel 01	%IW44	ARRAY [0..2] OF WORD		Read Holding Registry (Function Code 03)
		channel 01[0]	%IW44	WORD		Read #0515
		channel 01[1]	%IW45	WORD		Read #0516
		channel 01[2]	%IW46	WORD		Read #0517
		channel 01	%IX94.0	BOOL		Processing complete state
		channel 01	%QX32.0	BOOL		Trigger control bit

В нашем примере для записи выделены %QW13, %QW14 и %QW15. Для чтения выделены регистры %IW44, %IW45 и %IW46. Данные регистры можно напрямую использовать в программе для обмена данными с Ведомым:

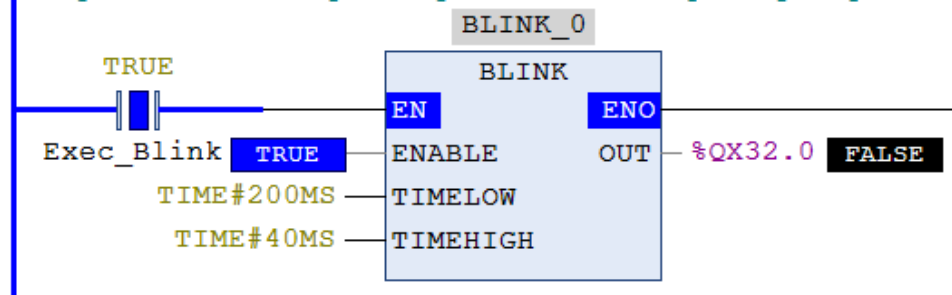
Отправка данных в регистры V3-V5 контроллера Optimus Drive AT16S0T-RU.



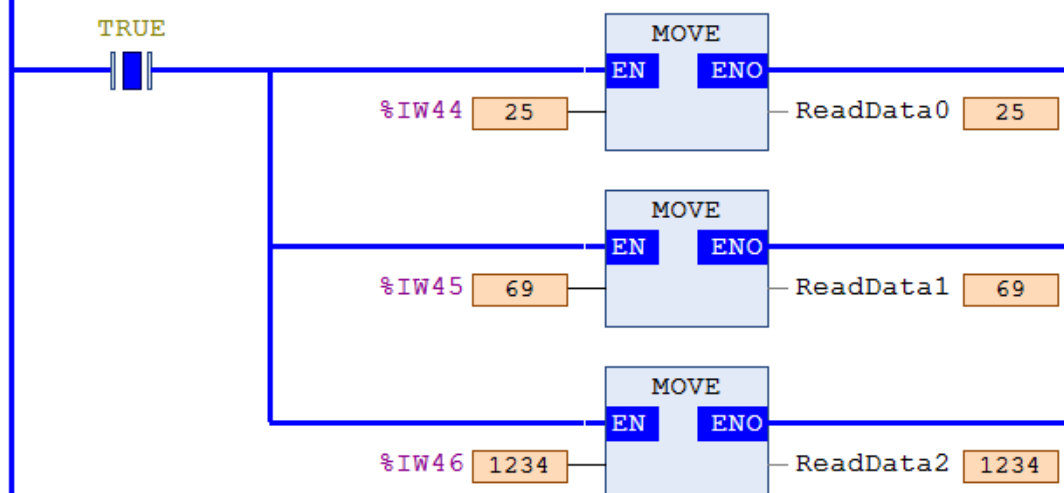
Для запуска триггера удобно использовать команду Blink:

	channel 01	%QX32.0	BOOL		Trigger control bit
--	------------	---------	------	--	---------------------

Запрос на чтение регистров V3-V5 контроллера Optimus Drive AT16S0T-RU.



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

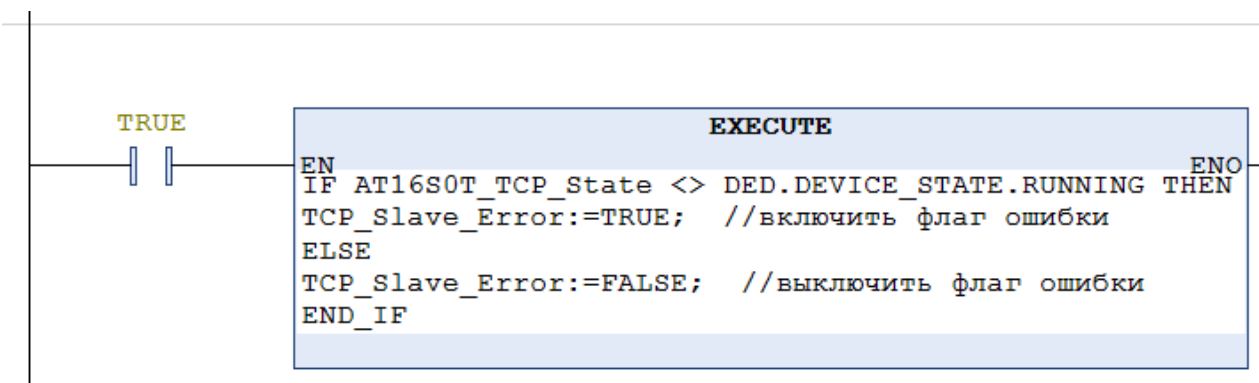


Ошибки связи можно проверить через свойство узла **GetDeviceState** и переменную типа **DED.DEVICE_STATE**

10 |  VAR **AT16S0T_TCP_State** DED.DEVICE_STATE

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

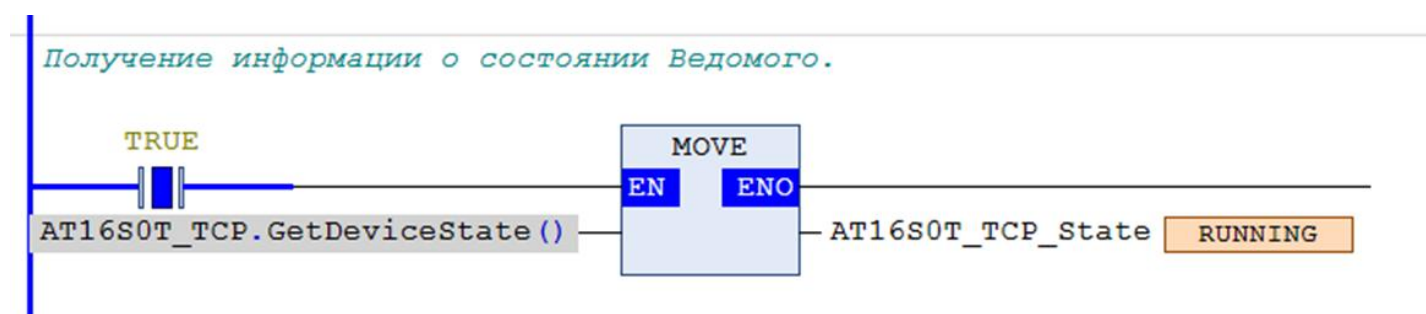
 VAR **TCP_Slave_Error** BOOL



Текст программы из блока Execute:

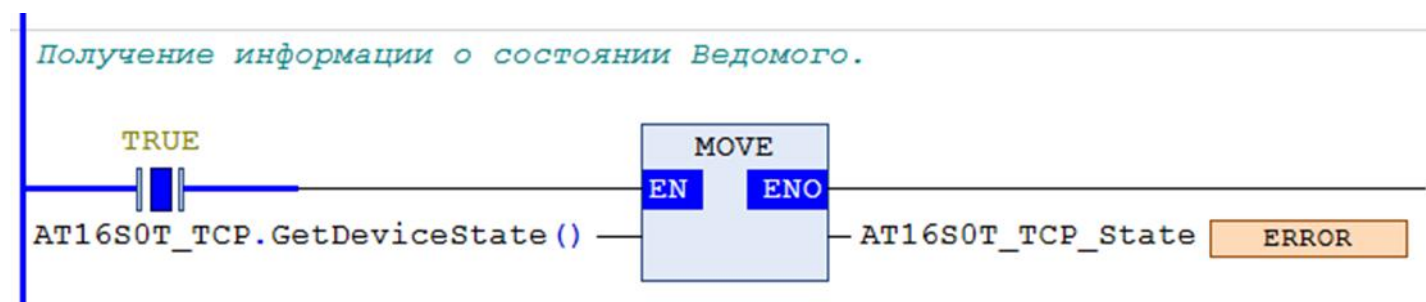
```
IF AT16S0T_TCP_State <> DED.DEVICE_STATE.RUNNING THEN
TCP_Slave_Error:=TRUE; //включить флаг ошибки
ELSE
TCP_Slave_Error:=FALSE; //выключить флаг ошибки
END_IF
```

Нормальное состояние ведомого:



Modbus_TCP.Slave_Error	Controller_1.Application	BOOL	FALSE
------------------------	--------------------------	------	-------

Потеря связи с ведомым:



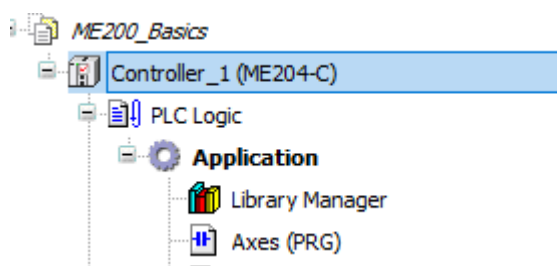
Modbus_TCP.Slave_Error	Controller_1.Application	BOOL	TRUE
------------------------	--------------------------	------	------

Связь по протоколу Modbus TCP Slave

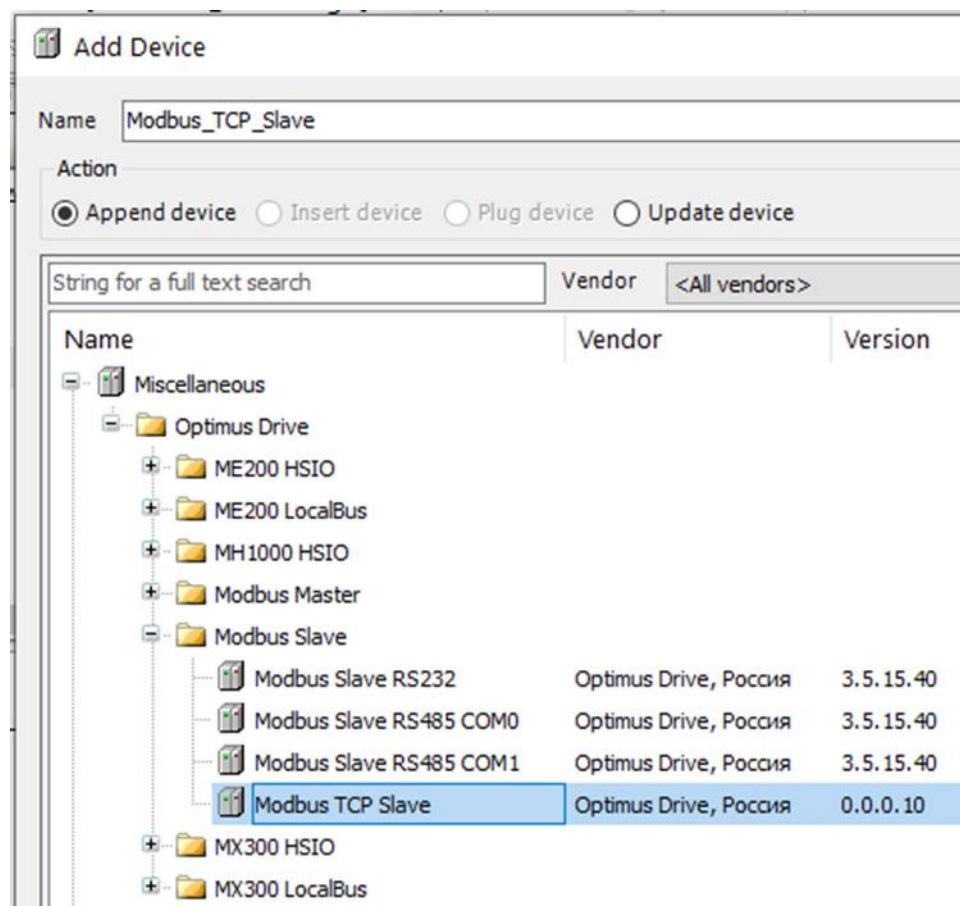
Контроллеры серии MH1048 могут работать как в режиме Modbus TCP Client (Master), так и в режиме Modbus TCP Server (Slave), причём оба режима могут использоваться одновременно. В данной главе рассматривается организация связи с контроллером по протоколу Modbus TCP Server (Slave).

Для организации связи с контроллером в режиме Modbus TCP Server в древо проекта необходимо добавить адаптер

Щёлкните правой кнопкой мышки на пункте **Device (MH1048)** и в отрывшемся меню выберите пункт **Add Device**.

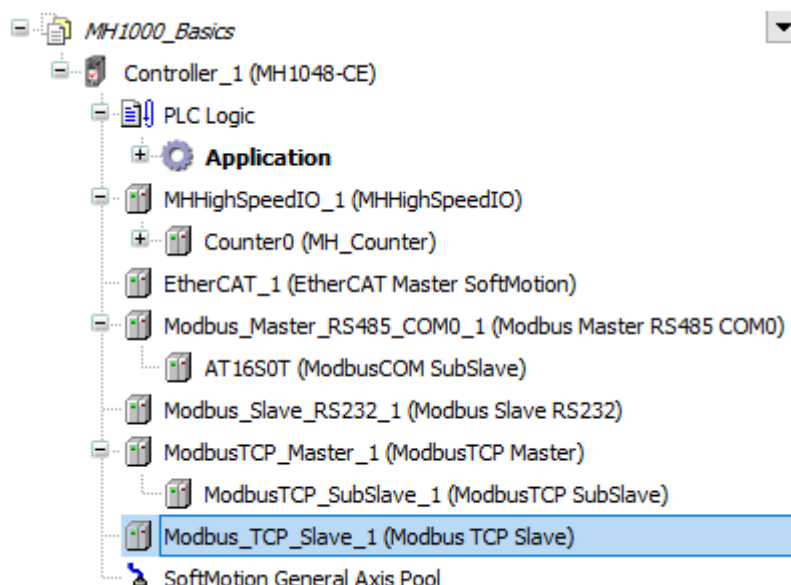


В открывшемся окне выберите пункт **Miscellaneous – Optimus Drive – Modbus Slave – Modbus TCP Slave**:



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

(идёт в древе проекта непосредственно от головного узла Device)



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 (начальный адрес 0x0000):

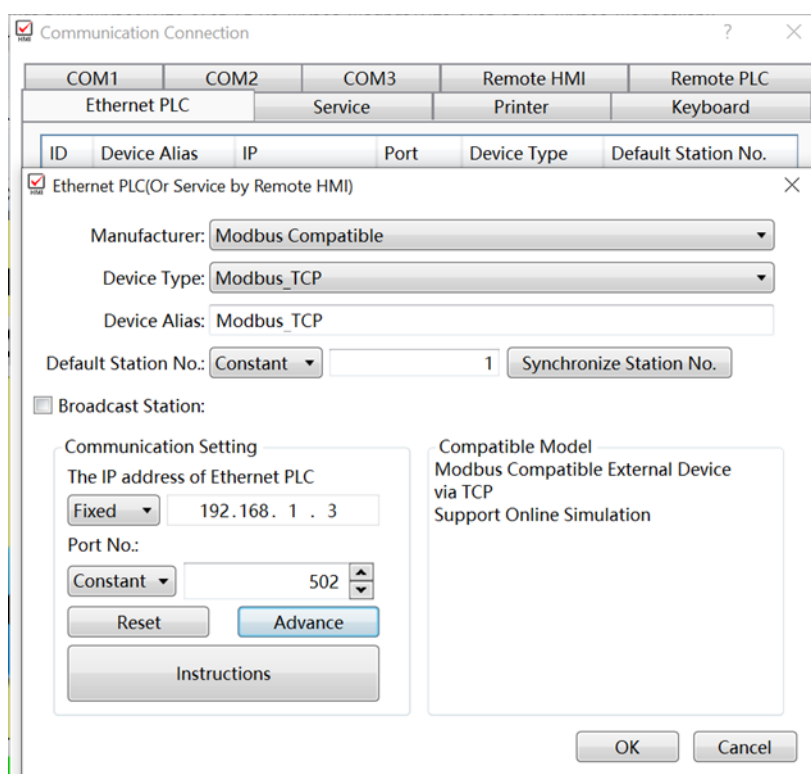
Тип	Диапазон	Функция	Инициализация	Количество
Q (выходы)	(QX0.0 ~ QX8191.7)	0x01,0x05,0x0f	0	65536
I (входы)	(IX0.0 ~ IX8191.7)	0x02	0	65536
M (данные)	%MW0~%MW65535	0x03,0x06,0x10	0	65536

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.

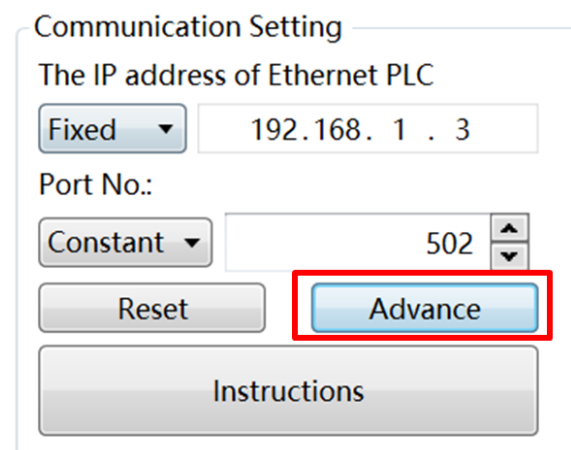
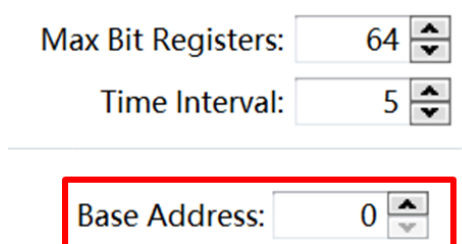
Например, маркеры %MX0.0.. %MX0.7 и %MX1.0.. %MX1.7 будут входить в состав регистра %MW0, маркеры %MX2.0.. %MX2.7 и %MX3.0.. %MX3.7 будут входить в состав регистра %MW1 и т.д.

Контроллеры серии MH1048 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.

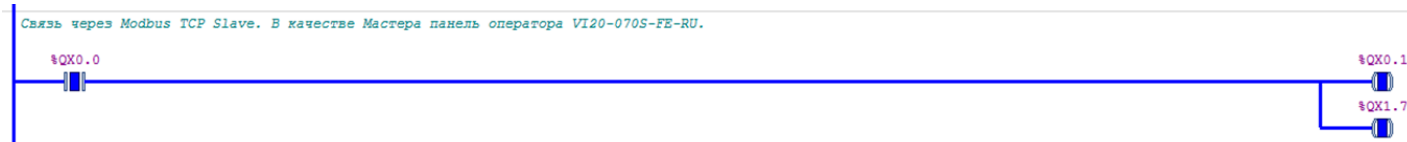
В панели необходимо выбрать стандартный Modbus TCP драйвер:



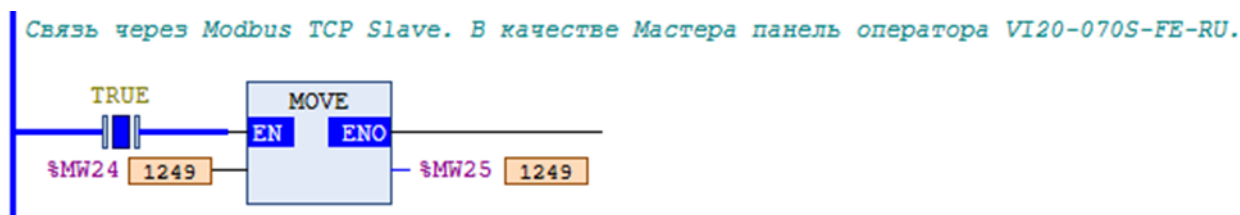
Указать IP адрес контроллера. В нашем примере 192.168.1.3 и во вкладке **Advance** убрать смещение адреса:

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

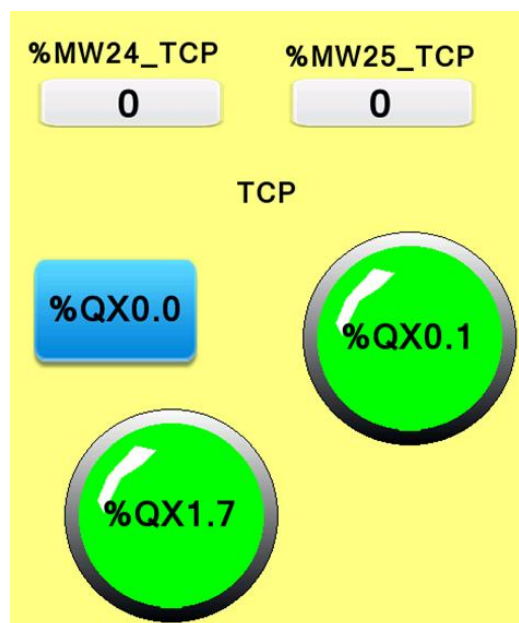


Регистр %QX0.0 - контакт, регистры %QX0.1 и %QX1.7 - выходные катушки.



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:

Device: Modbus_TCP:[Ethernet PLC:Modbus_TCP]

Station No: 1 ☐ Index

Address Type: 4X

Address: 24

а к регистру %MW25 так

Device: Modbus_TCP:[Ethernet PLC:Modbus_TCP]

Station No: 1 ☐ Index

Address Type: 4X

Address: 25

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

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

Format(Range):DDDDD(0~65534)

а к регистру %QX1.7 так:

☒ Standard Bit Address Input

☐ Use Address Tag

Device:

Station No: ☐ Index

☐ Bit-index within a Byte Register

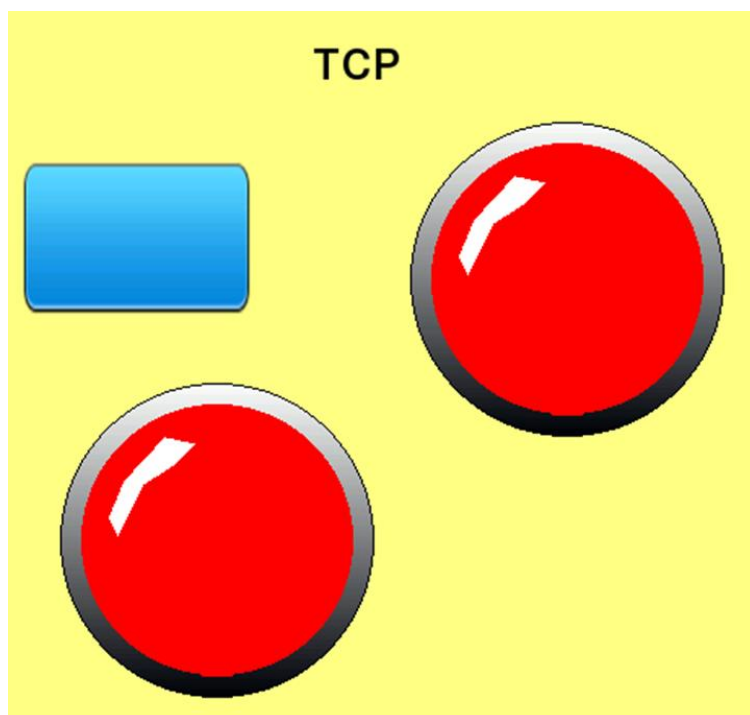
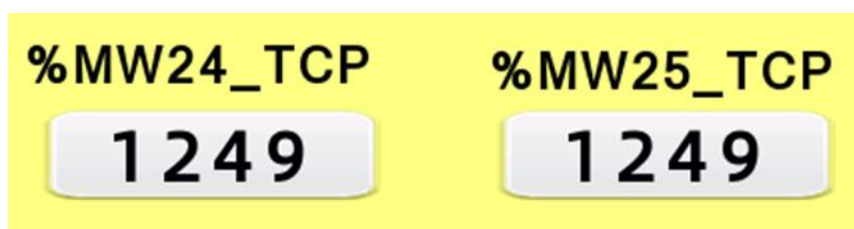
Address Type:

Address:

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 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 MH1048 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).

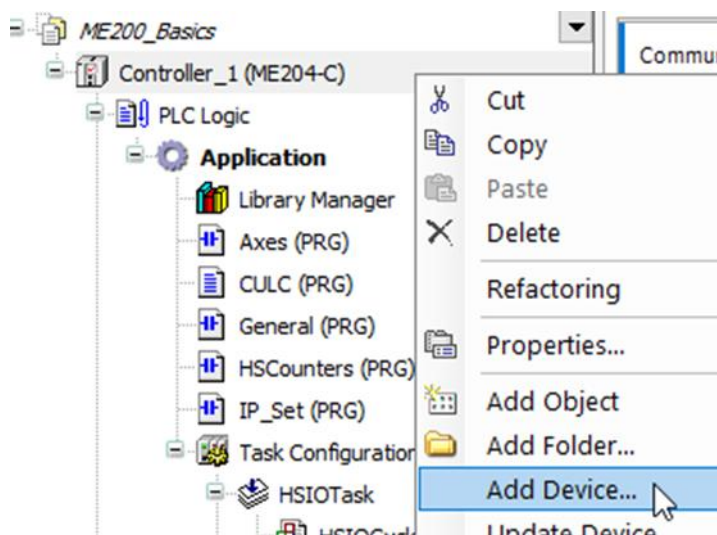
Для использования Ethernet/IP к проекту должны быть подключены следующие библиотеки:

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** (MH1048) и в отрывшемся меню выберите пункт **Add Device**



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

 Add Device

Name

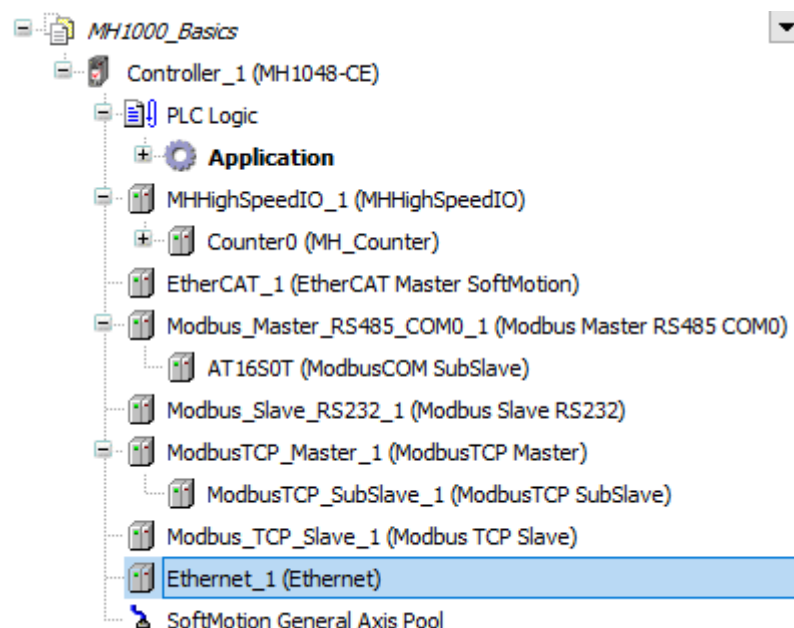
Action

☒ Append device ☐ Insert device ☐ Plug device ☐ Update device

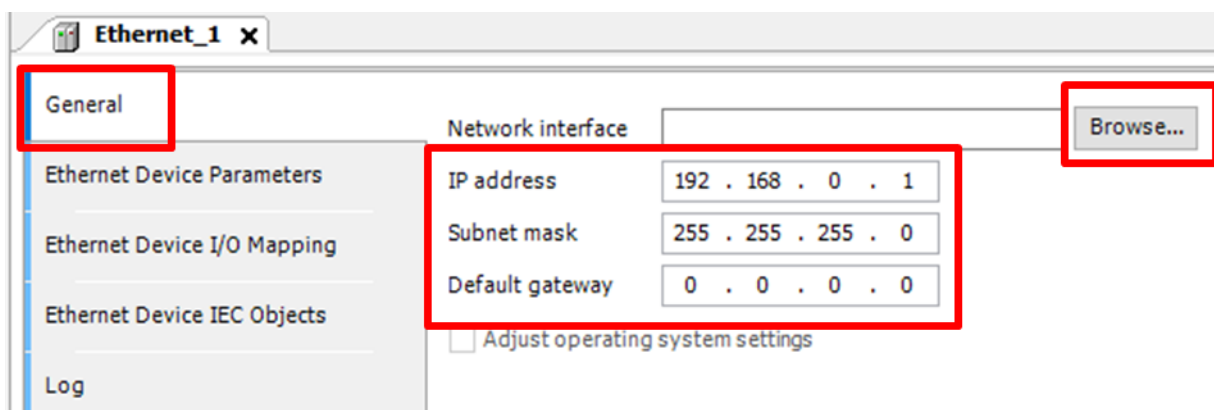
String for a full text search Vendor

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

В древе проекта появится пункт **Ethernet**:



Щёлкните дважды левой кнопкой мышки на пункте **Ethernet**, и в появившейся вкладке в разделе **General** необходимо сделать привязку порта к проекту:



Ethernet_1 X

General

Ethernet Device Parameters

Ethernet Device I/O Mapping

Ethernet Device IEC Objects

Log

Network interface

Browse...

IP address 192 . 168 . 0 . 1

Subnet mask 255 . 255 . 255 . 0

Default gateway 0 . 0 . 0 . 0

☐ Adjust operating system settings

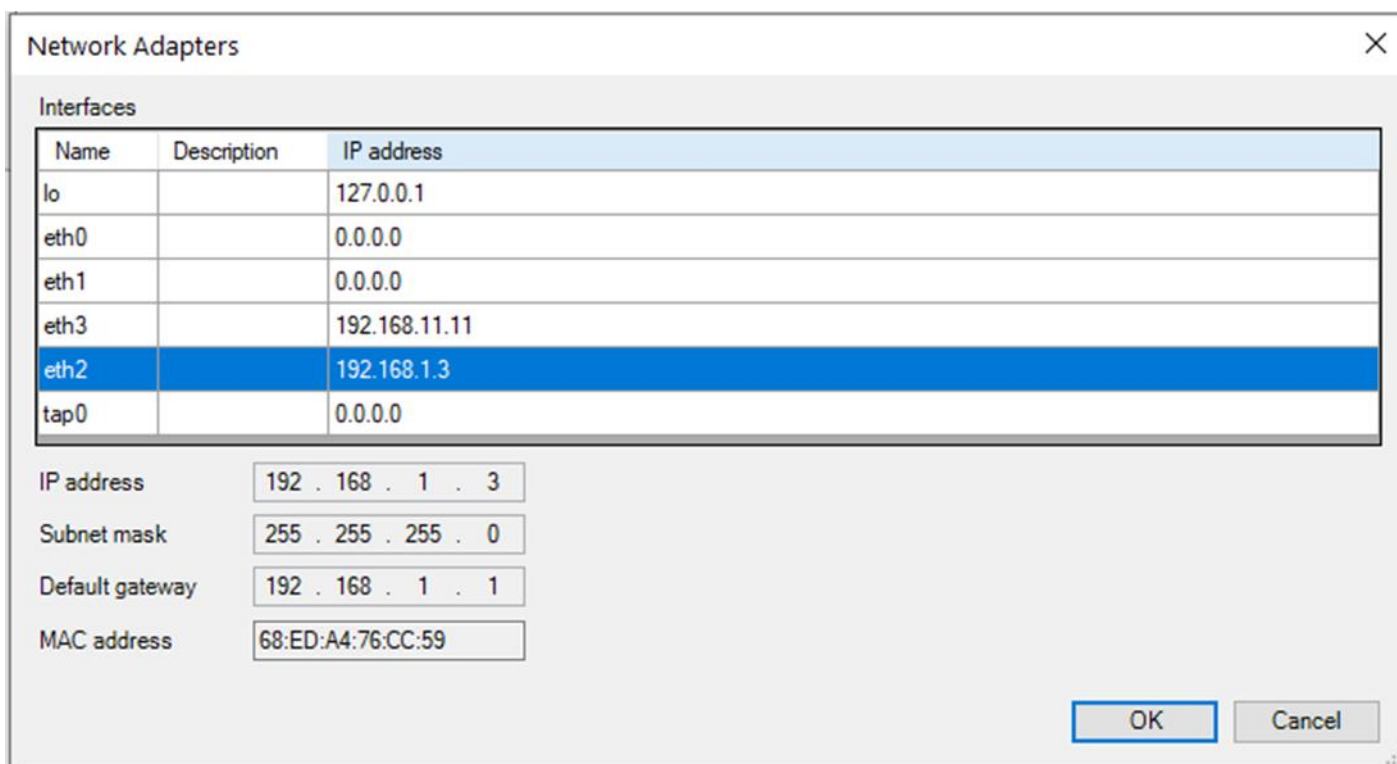
Задайте нужный IP адрес:

IP address 192 . 168 . 1 . 4

Subnet mask 255 . 255 . 255 . 0

Default gateway 192 . 168 . 1 . 1

и нажмите кнопку **Browse....** В появившемся окне выберите нужный интерфейс:



Network Adapters X

Interfaces

Name	Description	IP address
lo		127.0.0.1
eth0		0.0.0.0
eth1		0.0.0.0
eth3		192.168.11.11
eth2		192.168.1.3
tap0		0.0.0.0

IP address 192 . 168 . 1 . 3

Subnet mask 255 . 255 . 255 . 0

Default gateway 192 . 168 . 1 . 1

MAC address 68:ED:A4:76:CC:59

OK Cancel

Нажмите **OK**.

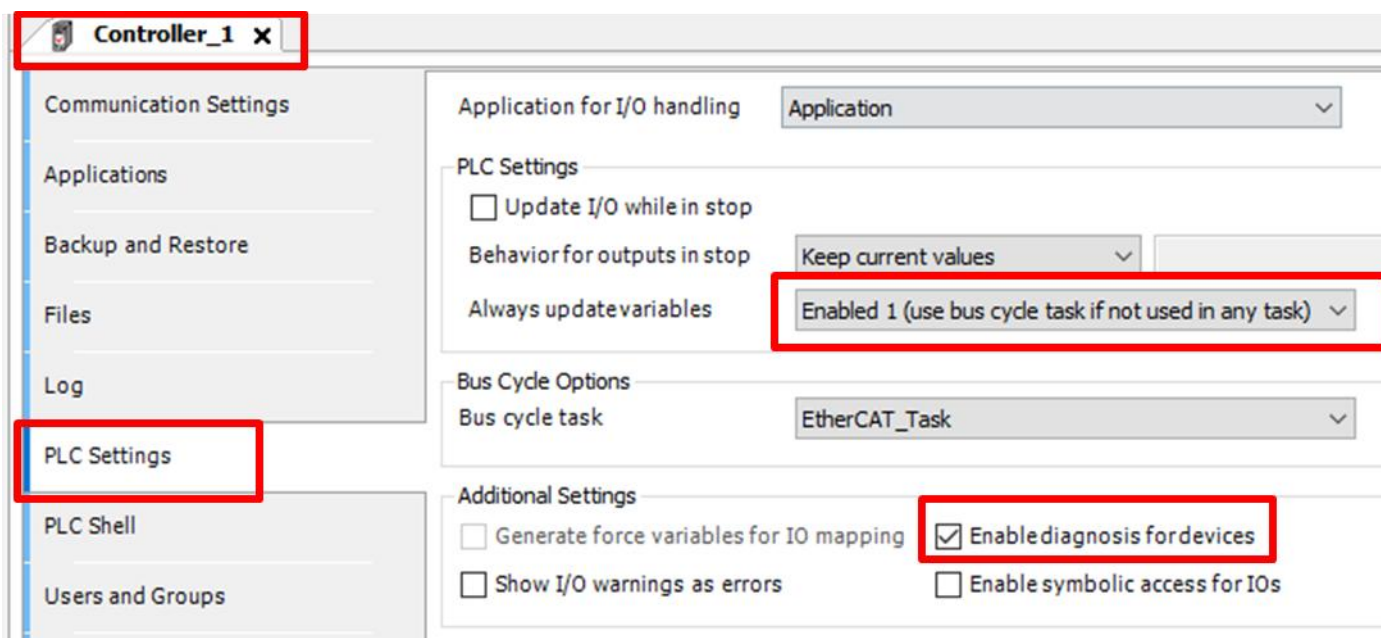
В поле Network Interface появится название интерфейса:

Network interface eth2 Browse...

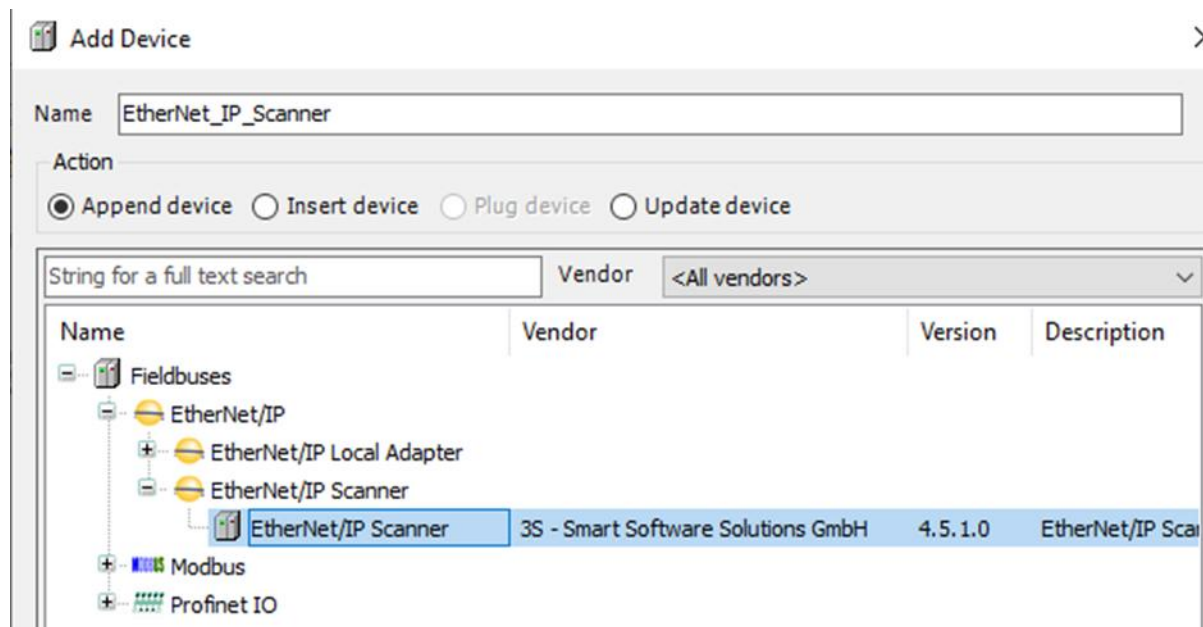
После чего Адаптер будет привязан к Ethernet порту с конкретным IP адресом.

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.

Также, для осуществления диагностики состояния подключенных к проекту устройств необходимо поставить флажок напротив пункта «**Enable diagnosis for devices**».



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**:

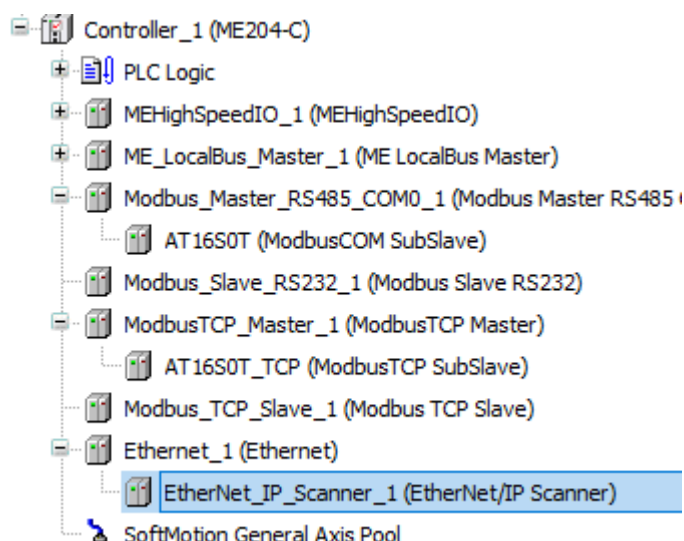


и нажмите **Add Device**:

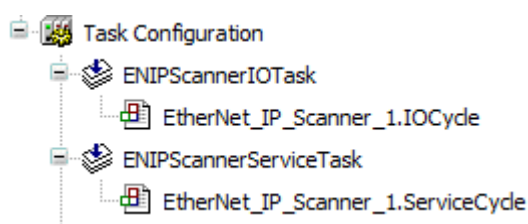
ow is open.)

Add Device

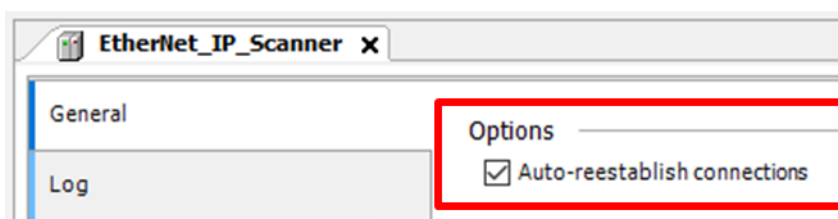
В древе проекта появится пункт **Ethernet/IP Scanner**:



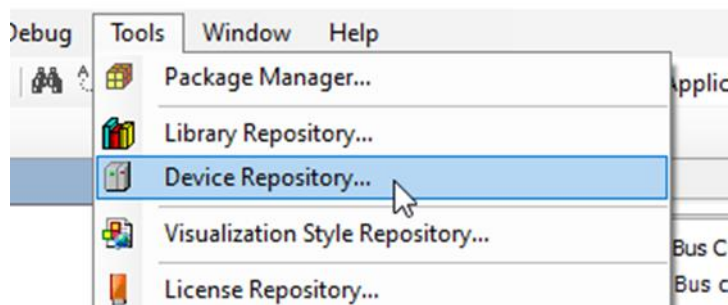
и автоматически будут созданы две задачи в Task Configuration, в которых будут обрабатываться все коммуникационные запросы протокола Ethernet/IP:



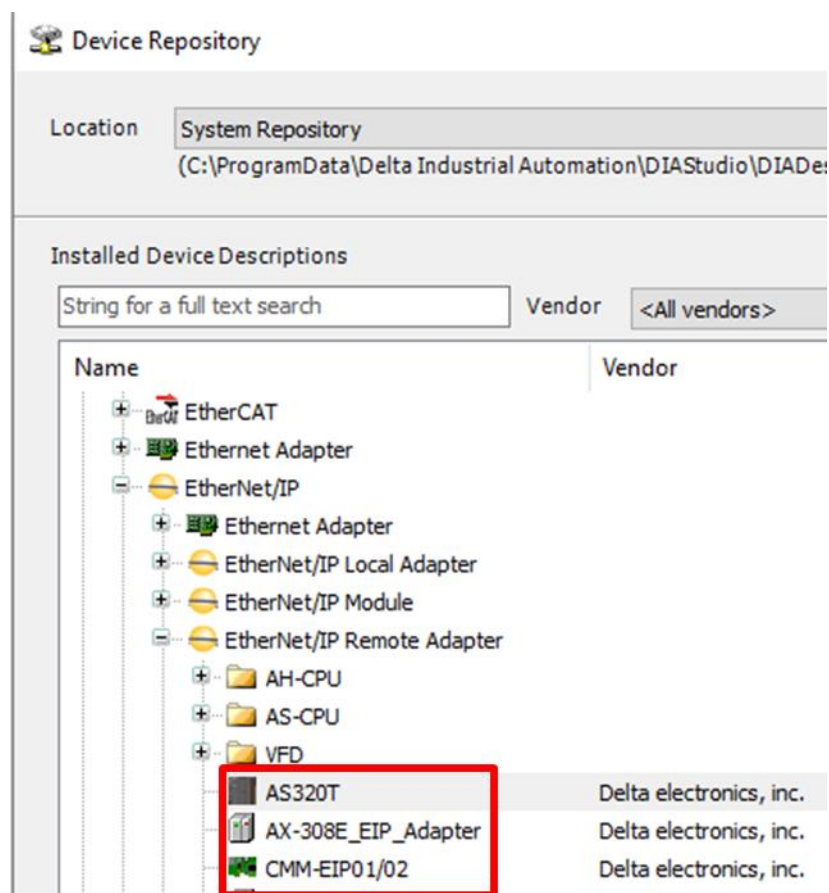
Щёлкните двойным щелчком левой кнопки мышки на пункте **Ethernet/IP Scanner** и в открывшейся вкладке выберите пункт General. Установите флажок автоматического восстановления соединения:



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

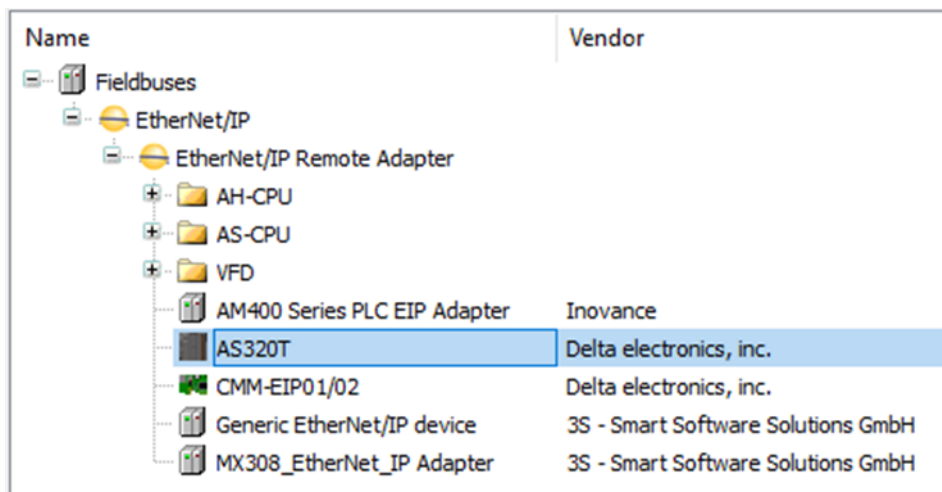


Ниже мы рассмотрим организацию связи с контроллером Delta AS320T и преобразователем частоты Delta MS300 посредством платы CMM-EIP02 по протоколу Ethernet/IP. Предполагается, что на этом этапе будут установлены соответствующие EDS файлы данных устройств:

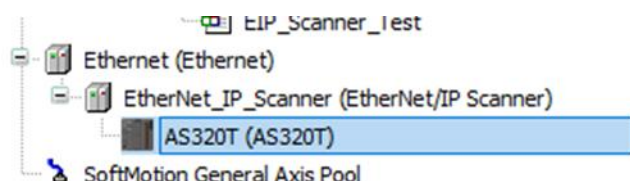


Пример настройки связи с контроллером Delta AS320T по протоколу Ethernet/IP

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 (MH1048), 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
Connection1						
Target Config data						
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
Assemblies					
User-Defined Parameters					
Log					
EtherNet/IP Parameters					
EtherNet/IP I/O Mapping					
Connection1			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 МН1048 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	Type	Value
%IW0	WORD	789
%IW99	WORD	1234
%QW0	WORD	0
%QW99	WORD	0
AS320T.eState	ADAPTERSTATE	RUNNING

Состояние связи с Ведомым можно проверить через свойство узла **eState**:

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

От Мастера к Ведомому:

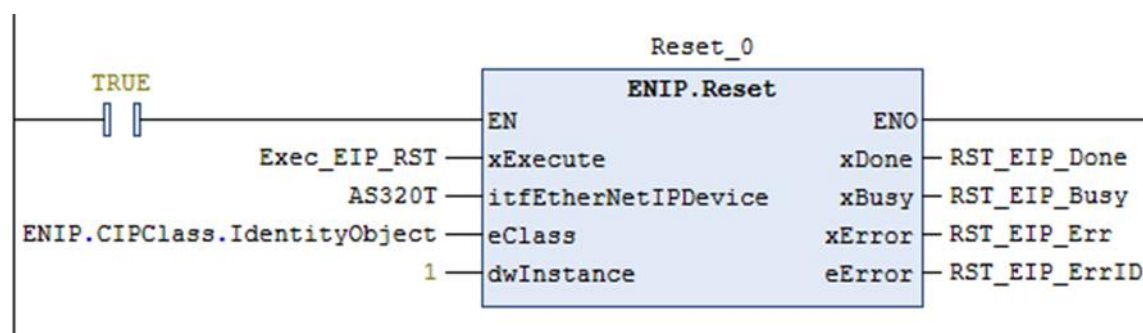
В программе Мастера задаются данные в регистрах %QW0 и %QW99

Watch 1		
Expression	Type	Value
%IW0	WORD	0
%IW99	WORD	0
%QW0	WORD	45
%QW99	WORD	555
 AS320T.eState	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**:

AS320T.xReset

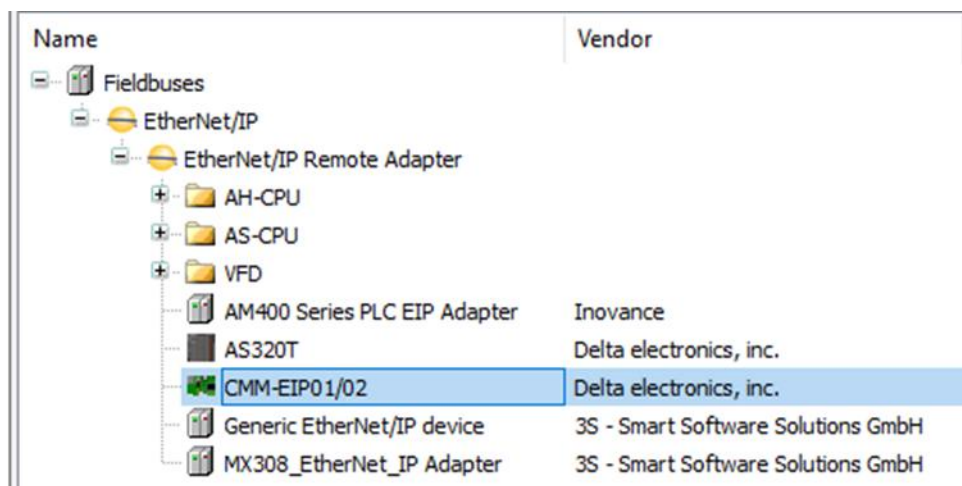

Пример настройки связи с ПЧ Delta MS300 по протоколу Ethernet/IP

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.

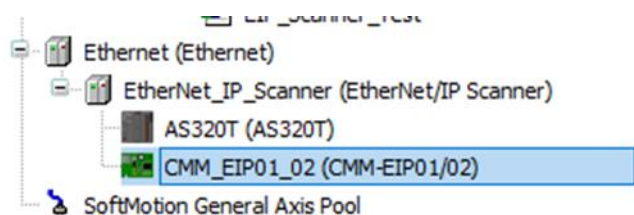
Также, необходимо убедиться, что код продукта в EDS файле на данную плату стоит 17157:

```
[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 =
```

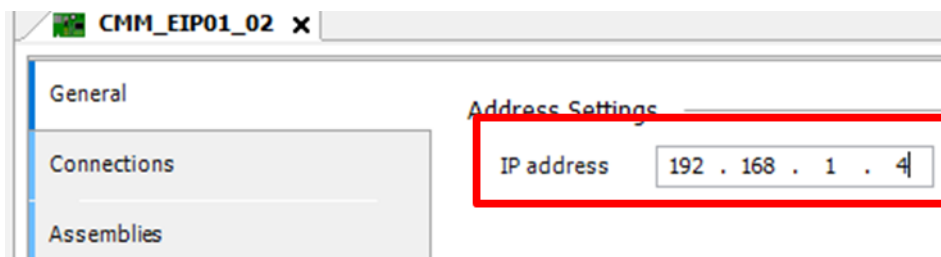
Для начала работы необходимо добавить Ведомое устройство в проект. Для этого щёлкните правой кнопкой мышки в древе проекта на пункте **Ethernet/IP Scanner** и в открывшемся меню выберите пункт **Add Device**. В открывшемся окне выберите из **Device Repository** устройство **CMM-EIP01/02**:



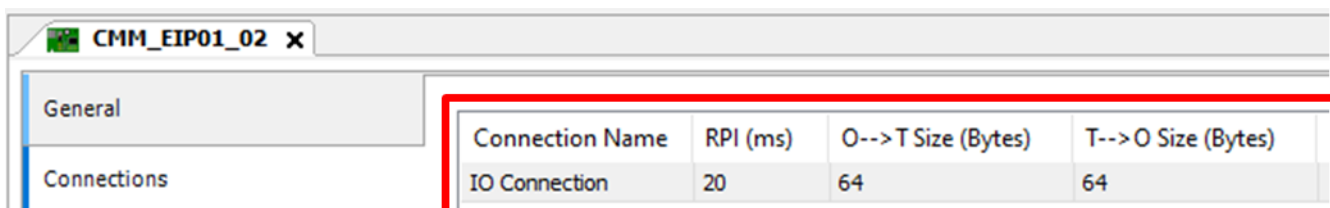
В древе проекта появится устройство:



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 приводу:

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 (MH1048) %QW100 - %QW131, которая будет автоматически сделана системой:

IO Connection					
		Error Code	%IW 100	UINT	
		Reserved2101	%IW 101	UINT	
		Frefuence Command	%IW 102	UINT	Hz
		Output Frequency	%IW 103	UINT	Hz
		Output Current	%IW 104	UINT	A
		DC Bus Voltage	%IW 105	UINT	
		Output Voltage	%IW 106	UINT	V
		Speed	%IW 107	UINT	
		Control	%QW 100	UINT	
		Frefuence Command	%QW 101	UINT	Hz
		Others	%QW 102	UINT	
		OUT_Value	%QW 103	UINT	
		OUT_Value	%QW 104	UINT	
		OUT_Value	%QW 105	UINT	
		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:

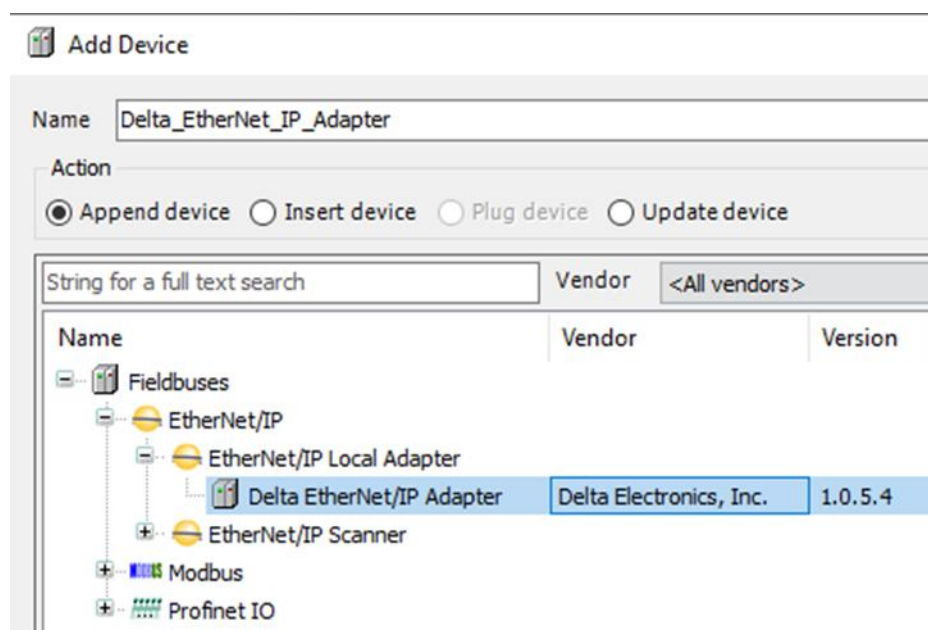
CMM_EIP01_02.eState	ADAPTERSTATE	RUNNING
CTRL_WORD	UINT	129
FREQ_CMD	UINT	2500
FREQ_SET	UINT	2500
OUT_FREQ	UINT	1154
DC_BUS	UINT	3060
STATUS_WORD	UINT	13571

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

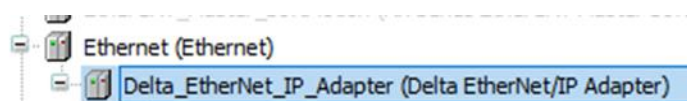
В среде программирования CODESYS есть специальная процедура для настройки связи между двумя контроллерами CODESYS по протоколу Ethernet/IP. В нашем примере в качестве Сканера (Мастера) будет выступать контроллер МН1048, а в качестве Адаптера (Ведомого) контроллер Delta AX-308E.

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

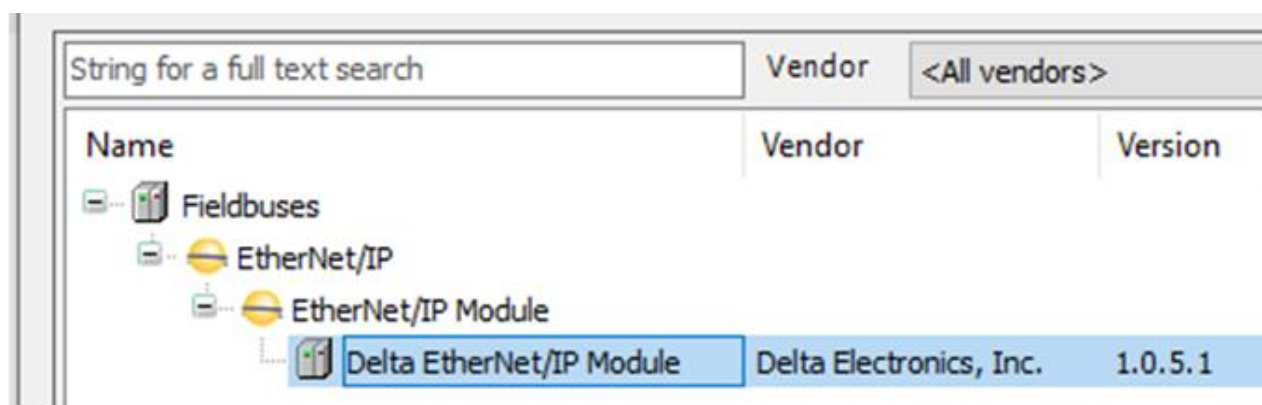
Добавьте в проект адаптер Ethernet. Выделите мышкой этот пункт в древе проекта и нажмите правую кнопку мышки. В появившемся меню выберите пункт **Add Device**. В открывшемся окне выберите **Delta Ethernet/IP Adapter**:



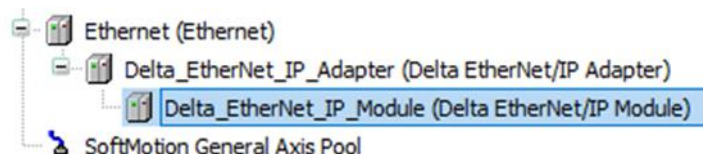
В древе проекта появится пункт:



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



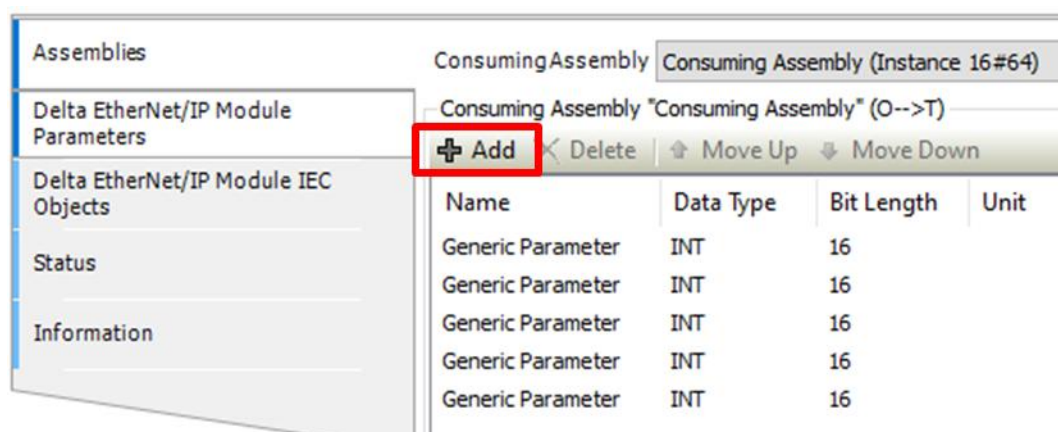
В древе проекта появится пункт:



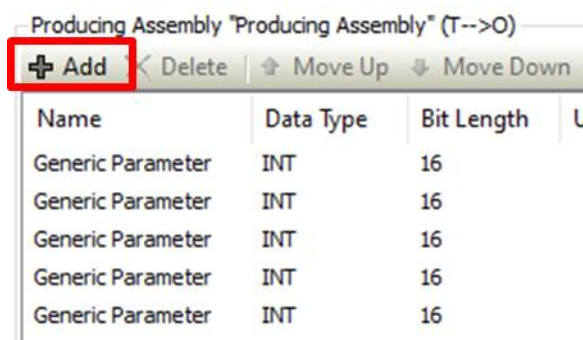
Щёлкните на этом пункте дважды левой кнопкой мышки и в открывшейся вкладке выберите раздел **Assemblies**:



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	
Delta EtherNet/IP Module Parameters	Variable	Mapping	Channel	Address Type Unit
Delta EtherNet/IP Module I/O Mapping			Generic Parameter	%IW2 INT
Delta EtherNet/IP Module IEC Objects			Generic Parameter	%IW3 INT
Status			Generic Parameter	%IW4 INT
Information			Generic Parameter	%IW5 INT
			Generic Parameter	%IW6 INT
			Generic Parameter	%QW1 INT
			Generic Parameter	%QW2 INT
			Generic Parameter	%QW3 INT
			Generic Parameter	%QW4 INT
			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**:

Delta_EtherNet_IP_Adapter X

General

Tags

Log

Delta EtherNet/IP Adapter Parameters

Delta EtherNet/IP Adapter I/O Mapping

Delta EtherNet/IP Adapter IEC Objects

Status

Information

EDS File

Vendor name

Delta electronics, inc.

Vendor ID

799

Product name

AX-308EA0MA1T

Product code

16386

Major revision

1

Minor revision

1

Support ACD

☒

Enable ACD

☐

Enable LLDP

☒

Install to Device Repository...

Export EDS File...

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.

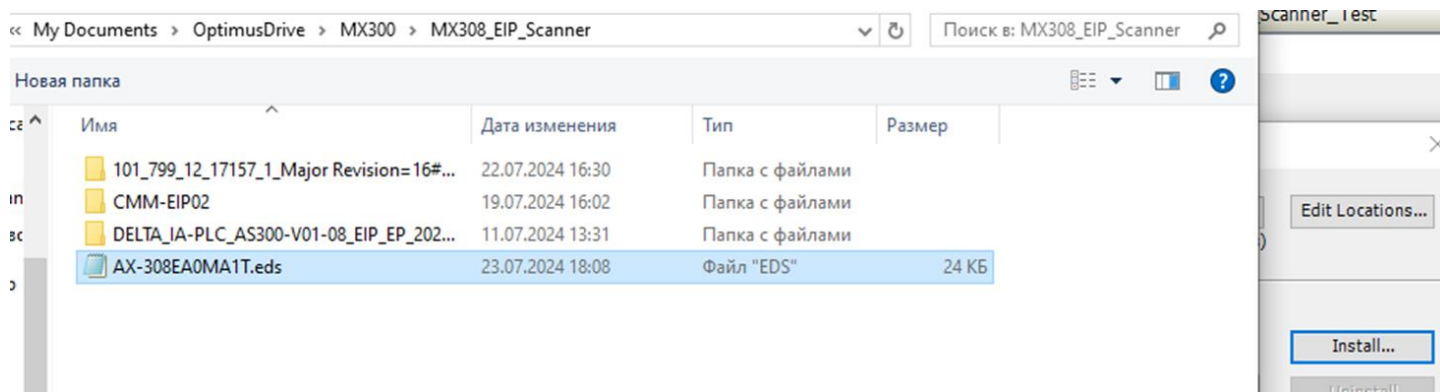
Для изменения поля откройте файл в Блокноте (Notepad) и измените название поле **ProdName**

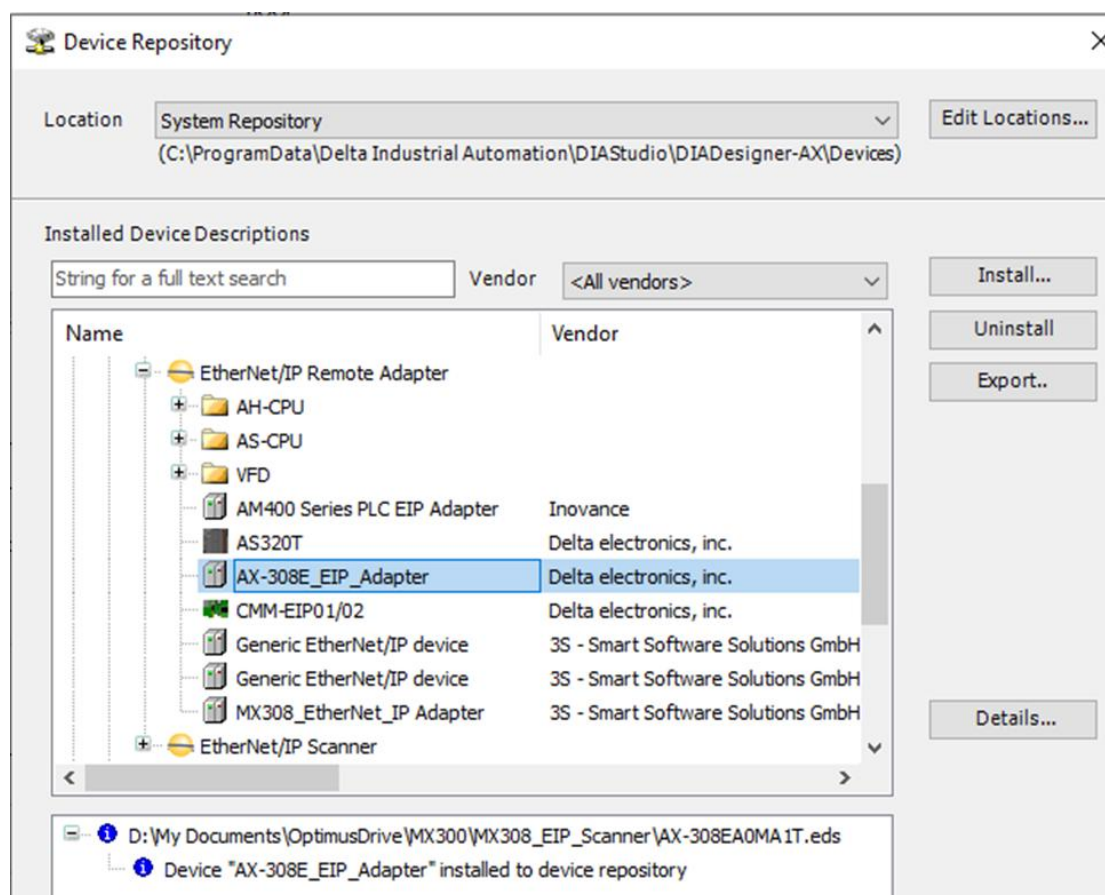
```
[Device]
VendCode = 799;
VendName = "Delta electronics, inc.";
ProdType = 12;
ProdTypeStr = "Communications Adapter";
ProdCode = 16386;
MajRev = 1;
MinRev = 1;
ProdName = "AX-308EA0MA1T";
```

например на такое:

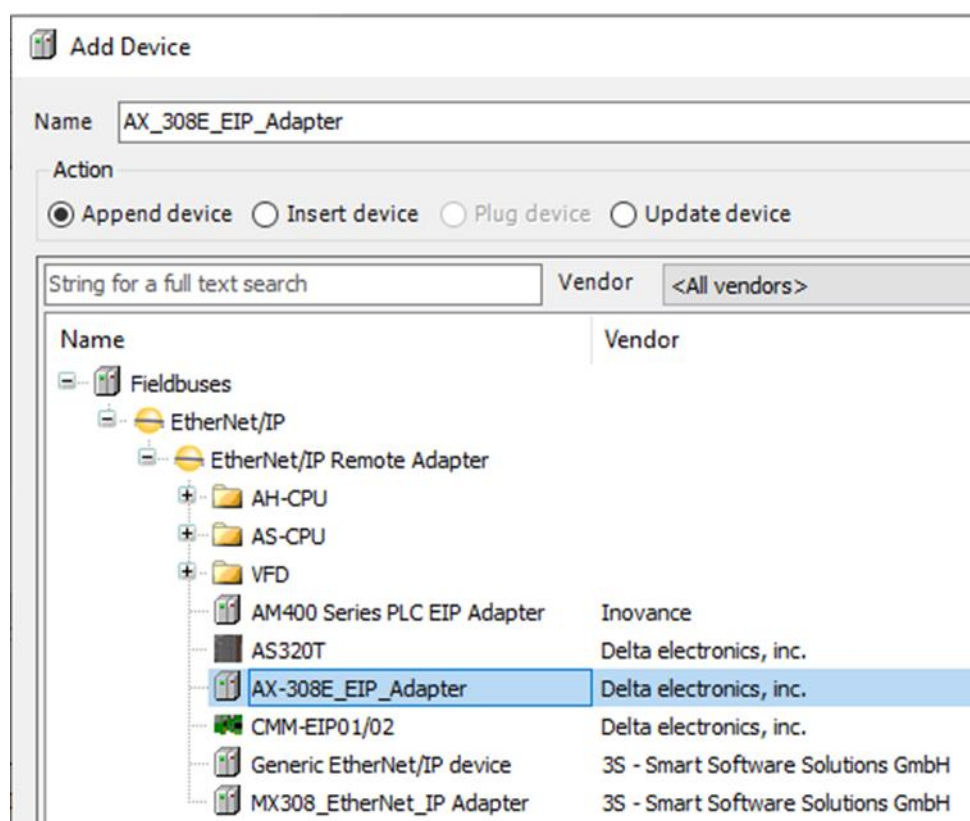
```
[Device]
VendCode = 799;
VendName = "Delta electronics, inc.";
ProdType = 12;
ProdTypeStr = "Communications Adapter";
ProdCode = 16386;
MajRev = 1;
MinRev = 1;
ProdName = "AX-308E_EIP_Adapter";
```

Далее файл необходимо импортировать в репозиторий устройств Мастера (МН1000). Т.е. там появится устройство типа Adapter Ethernet/IP. Данная процедура, как и создание проекта для контроллеров МН1000, описаны в соответствующих Главах настоящего Руководства.

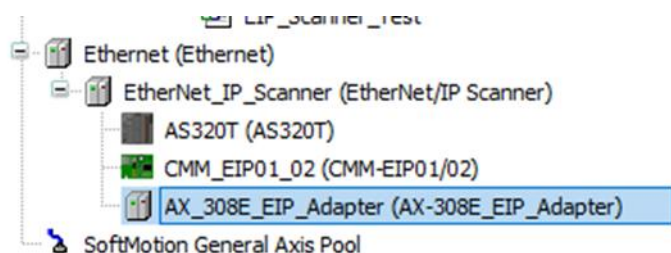




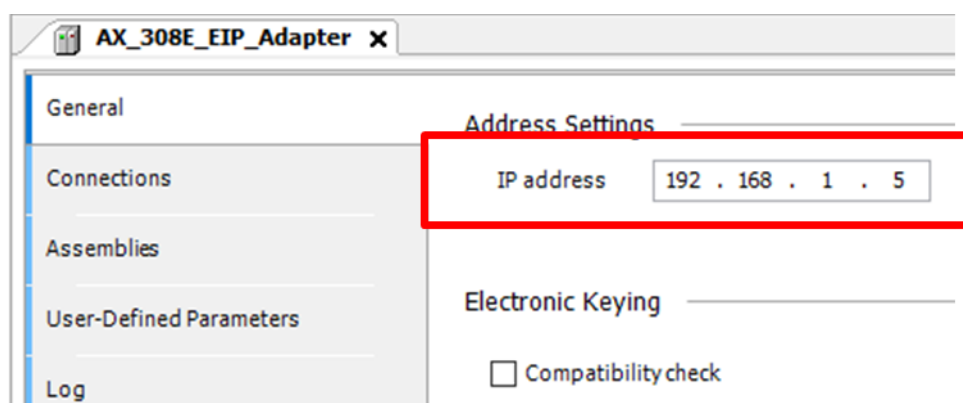
Добавьте адаптер в проект Мастера:



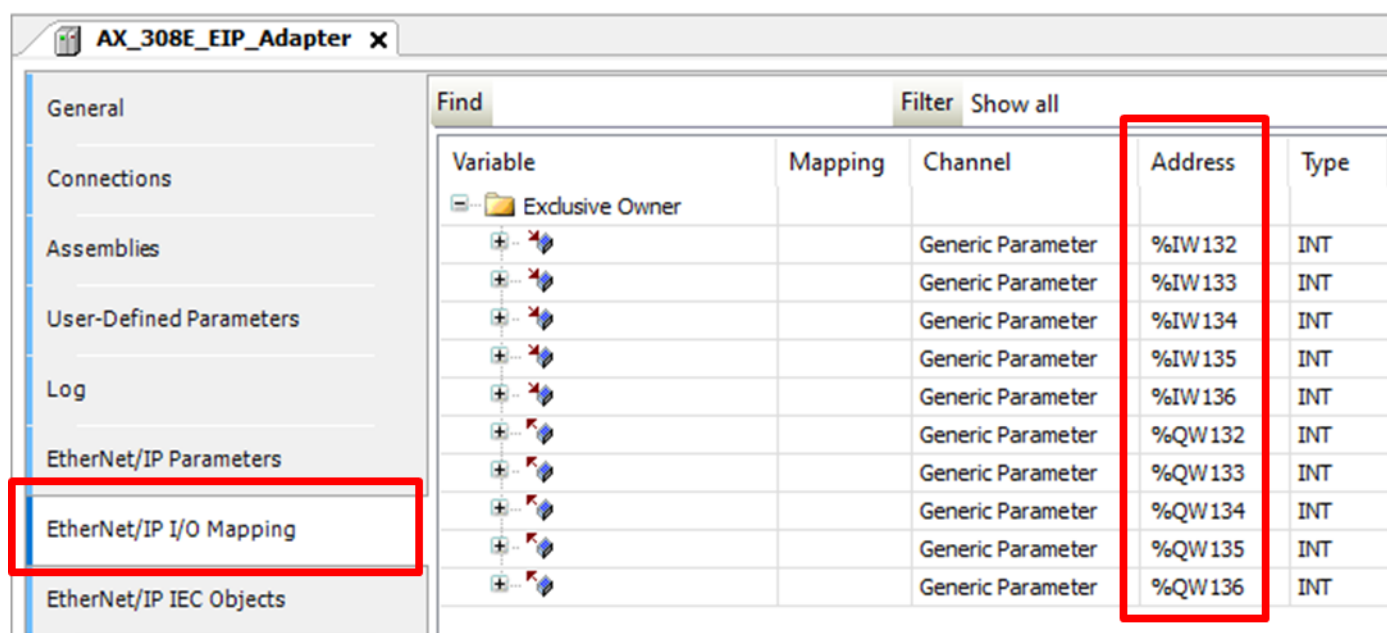
В древе проекта появится устройство:



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 МН1048) в регистре %QW132 задать число 45:

%IW132	WORD	0
%QW132	WORD	45
AX_308E_EIP_Adapter.eState	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 (МН1048) %IW132 появится число 555:

%IW132	WORD	555
%QW132	WORD	0
AX_308E_EIP_Adapter.eState	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 MH1048 могут работать как в режиме **Ethernet/IP Scanner** (Master), так и в режиме **Ethernet/IP Adapter** (Slave). В данной главе рассматривается организация связи контроллера MH1048 с Мастером в режиме Ethernet/IP Adapter (Slave).

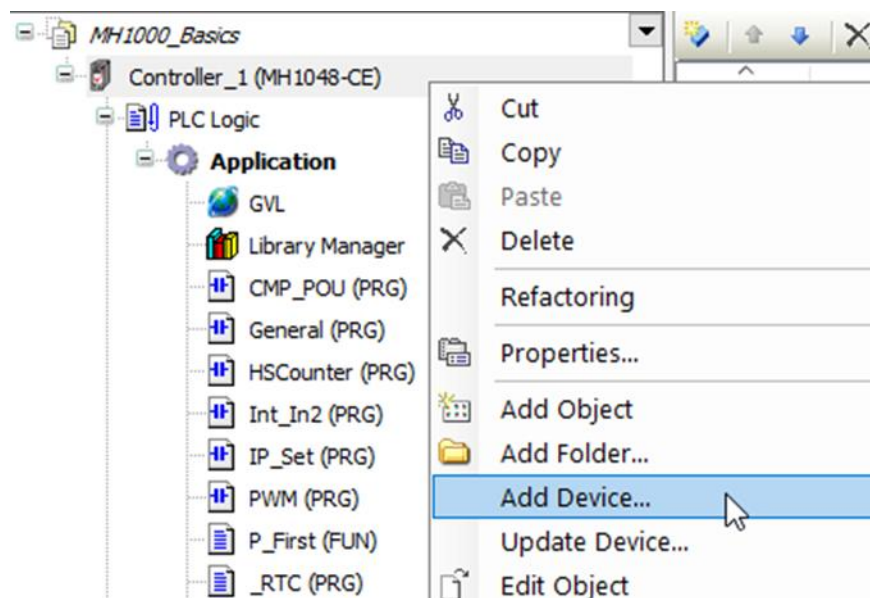
Для использования Ethernet/IP к проекту должны быть подключены следующие библиотеки:

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** (MH1048) и в отрывшемся меню выберите пункт **Add Device**



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

 Add Device

Name

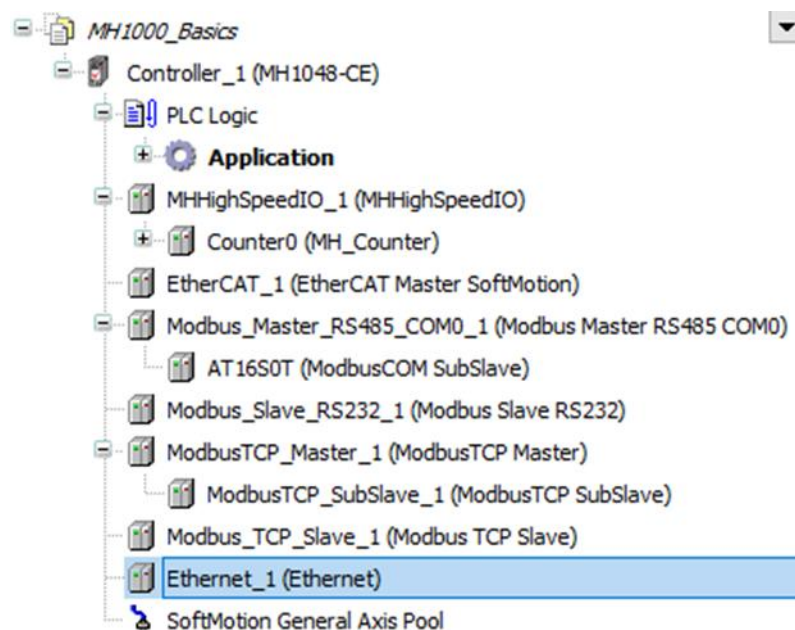
Action

☒ Append device ☐ Insert device ☐ Plug device ☐ Update device

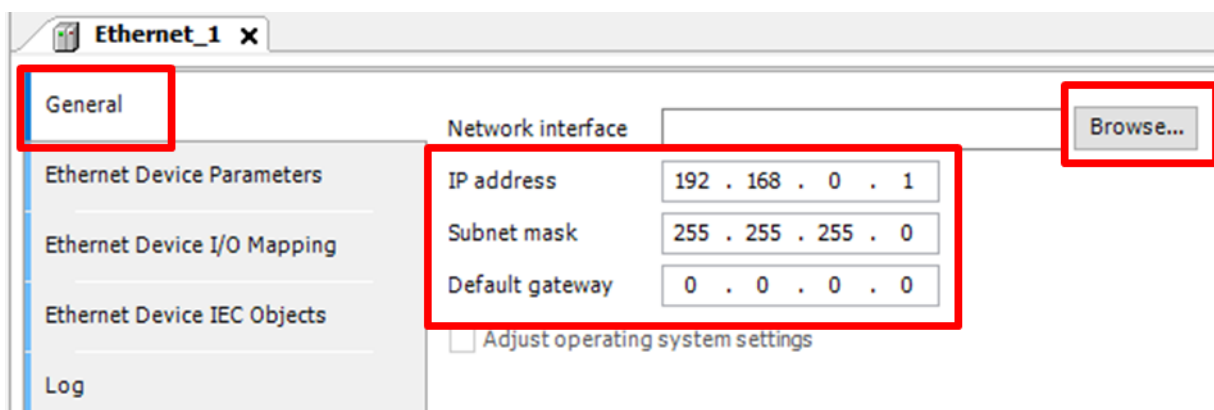
String for a full text search Vendor

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

В древе проекта появится пункт **Ethernet**:



Щёлкните дважды левой кнопкой мышки на пункте **Ethernet**, и в появившейся вкладке в разделе **General** необходимо сделать привязку порта к проекту:



Ethernet_1 X

General

Ethernet Device Parameters

Ethernet Device I/O Mapping

Ethernet Device IEC Objects

Log

Network interface

Browse...

IP address 192 . 168 . 0 . 1

Subnet mask 255 . 255 . 255 . 0

Default gateway 0 . 0 . 0 . 0

☐ Adjust operating system settings

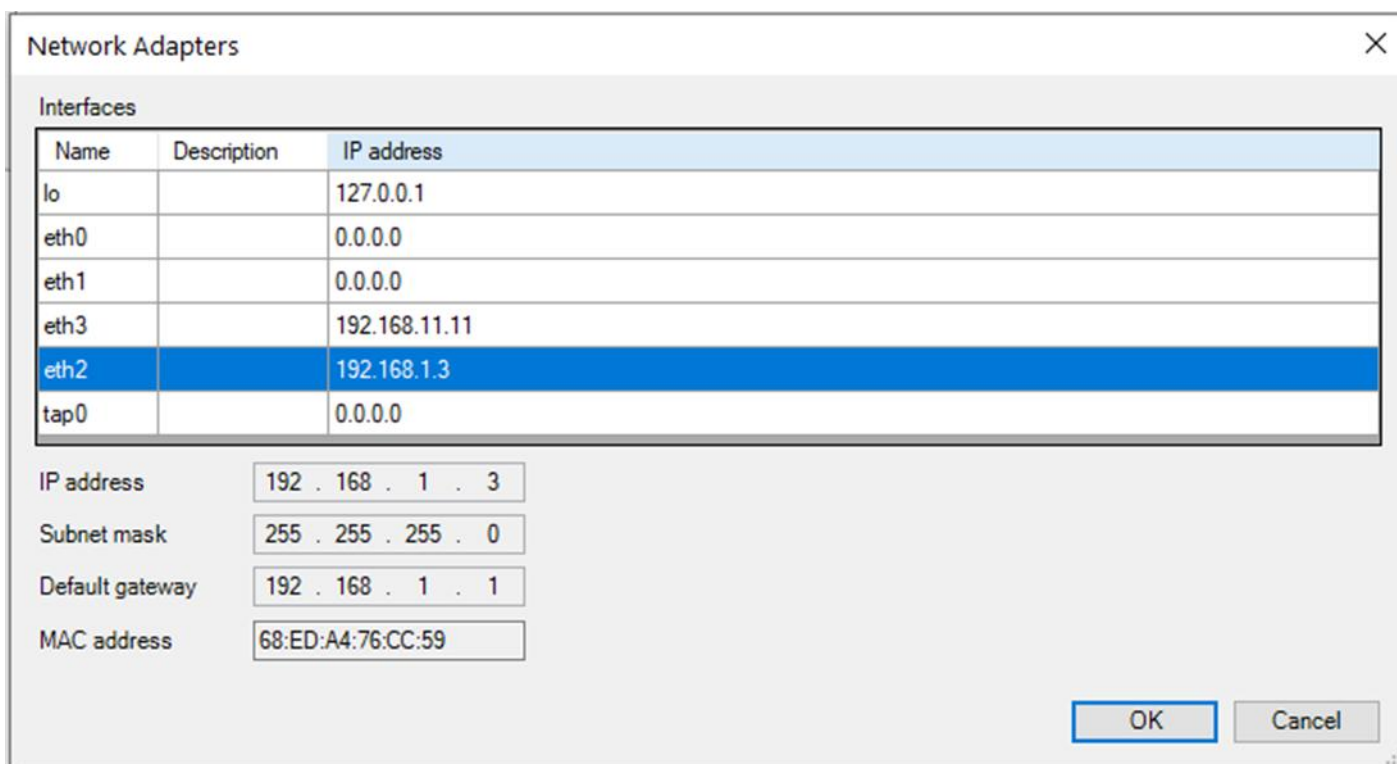
Задайте нужный IP адрес:

IP address 192 . 168 . 1 . 3

Subnet mask 255 . 255 . 255 . 0

Default gateway 192 . 168 . 1 . 1

и нажмите кнопку **Browse....** В появившемся окне выберите нужный интерфейс:



Network Adapters X

Interfaces

Name	Description	IP address
lo		127.0.0.1
eth0		0.0.0.0
eth1		0.0.0.0
eth3		192.168.11.11
eth2		192.168.1.3
tap0		0.0.0.0

IP address 192 . 168 . 1 . 3

Subnet mask 255 . 255 . 255 . 0

Default gateway 192 . 168 . 1 . 1

MAC address 68:ED:A4:76:CC:59

OK Cancel

Нажмите **OK**.

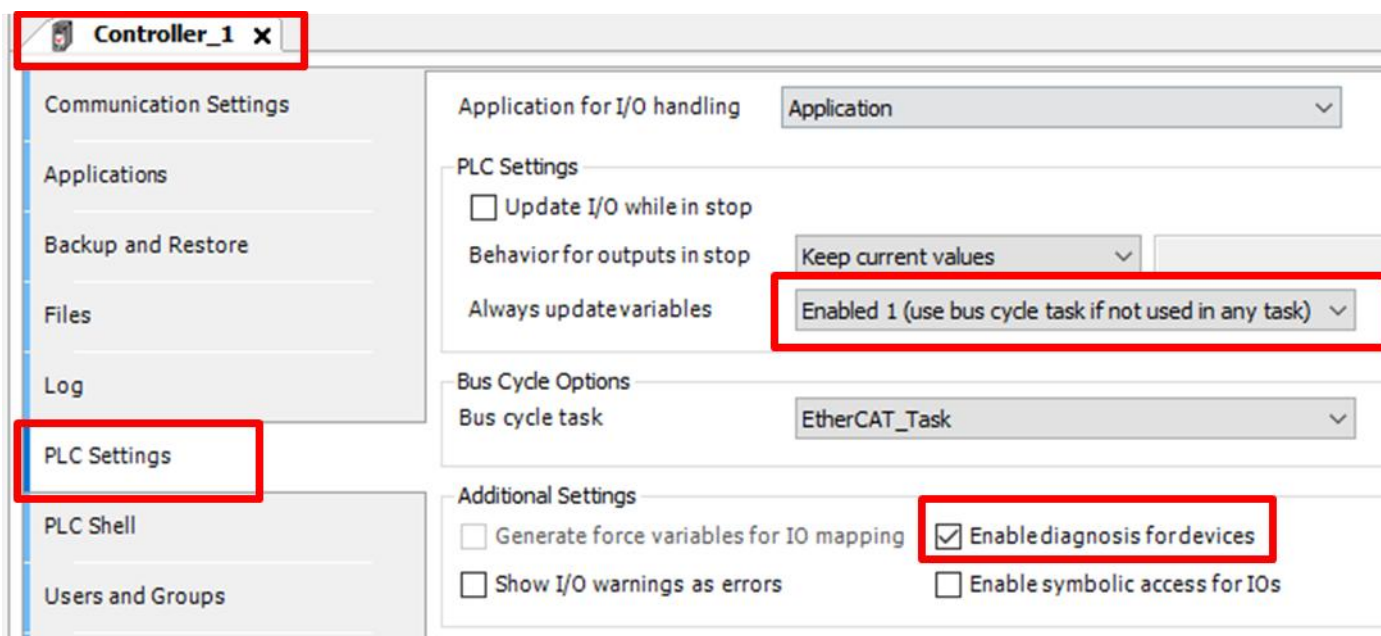
В поле Network Interface появится название интерфейса:

Network interface eth2 Browse...

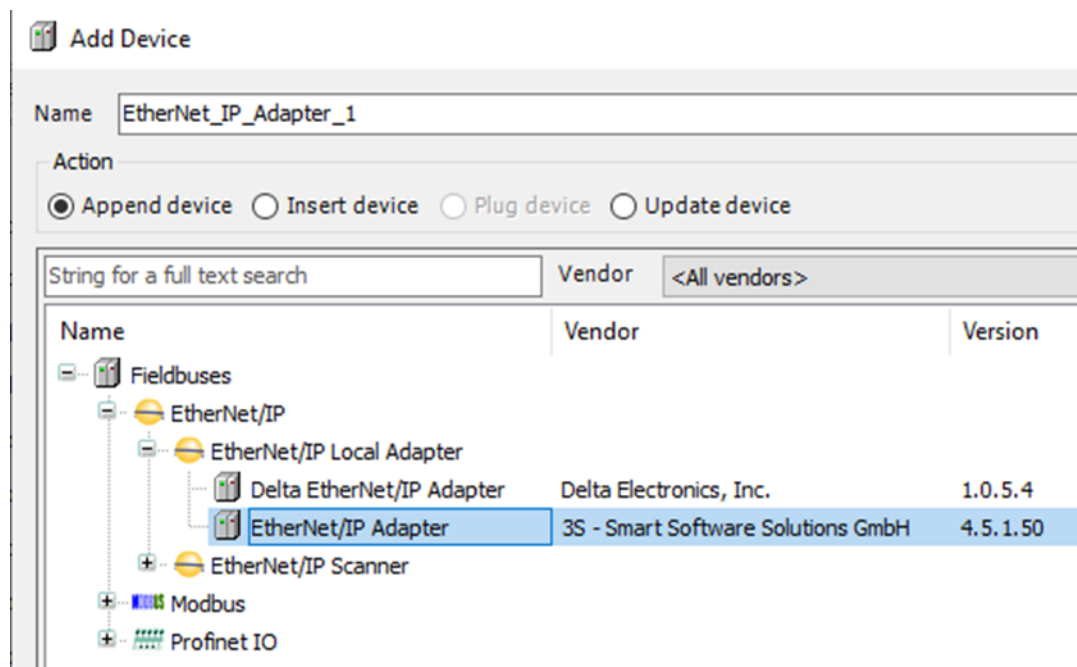
После чего Адаптер будет привязан к Ethernet порту с конкретным IP адресом.

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.

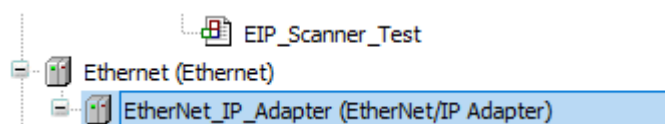
Также, для осуществления диагностики состояния подключенных к проекту устройств необходимо поставить флажок напротив пункта «Enable diagnosis for devices».



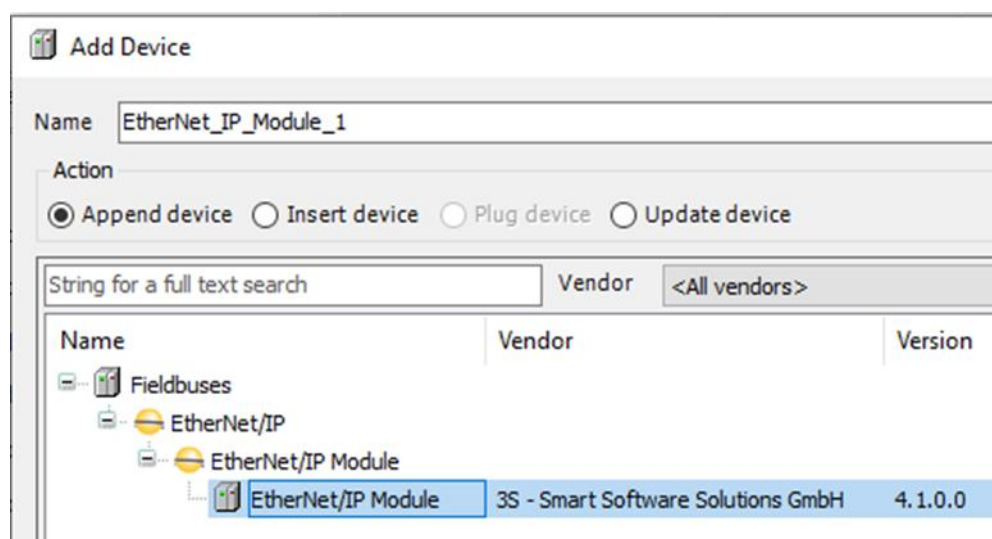
Щёлкните в древе правой кнопкой мышки на пункте **Ethernet** и в открывшемся меню выберите пункт **Add Device**. Далее в открывшемся окне выберите пункт **Fieldbuses – Ethernet/IP Adapter**:



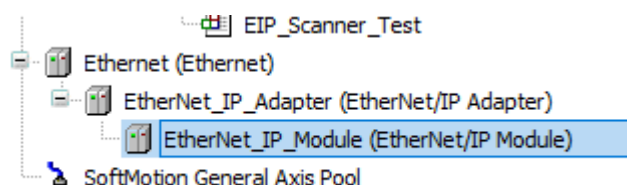
В древе проекта появится устройство:



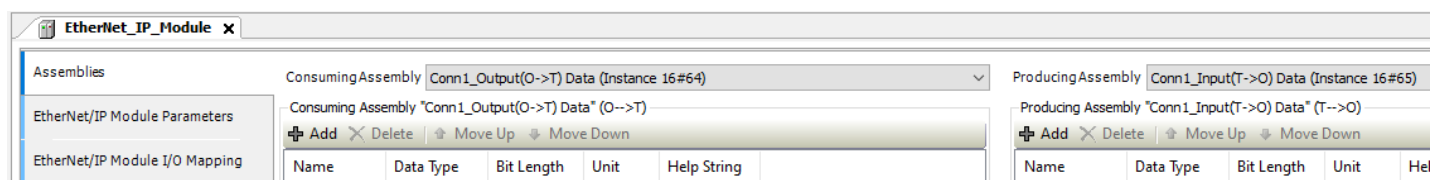
Щёлкните правой кнопкой мышки на данном пункте и добавьте устройство типа **Ethernet/IP Module**:



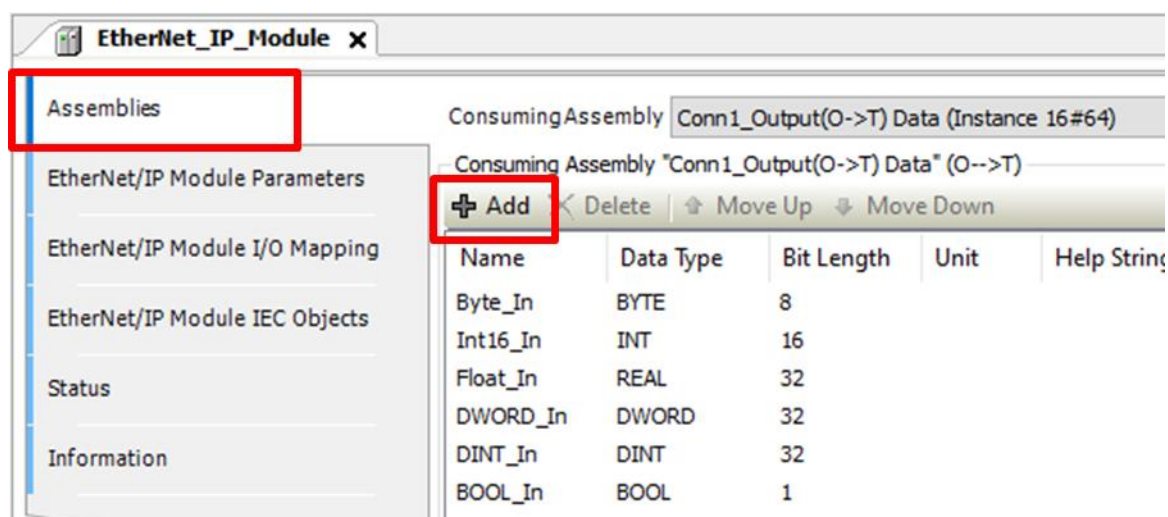
В древе проекта появится устройство:



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

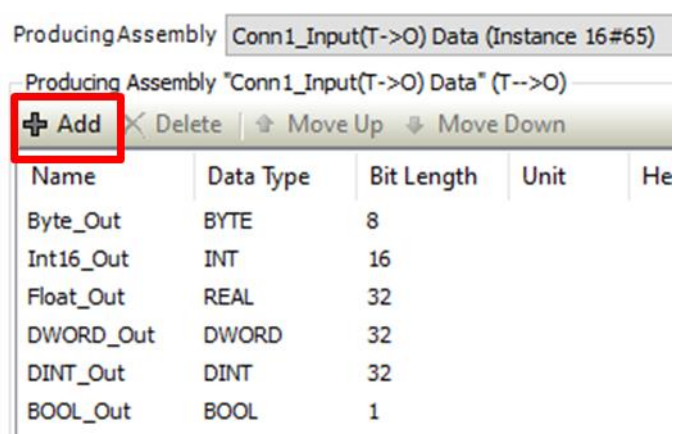


В данном разделе создаются пакеты для отправки и приёма от Мастера. Для добавления регистров в пакет на приём от Мастера нажимайте кнопку **Add** в поле **Consuming Assembly**:

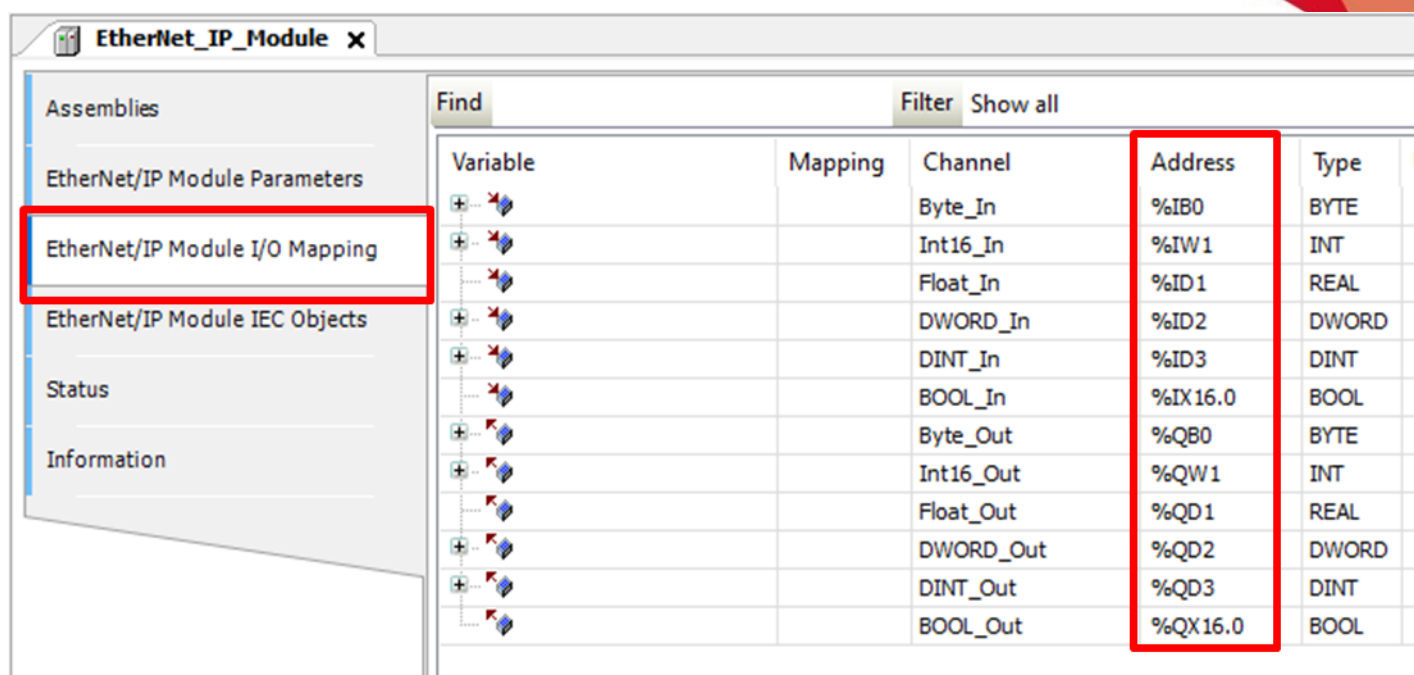


Теги можно создавать сразу с названием и любого из поддерживаемых типов данных.

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

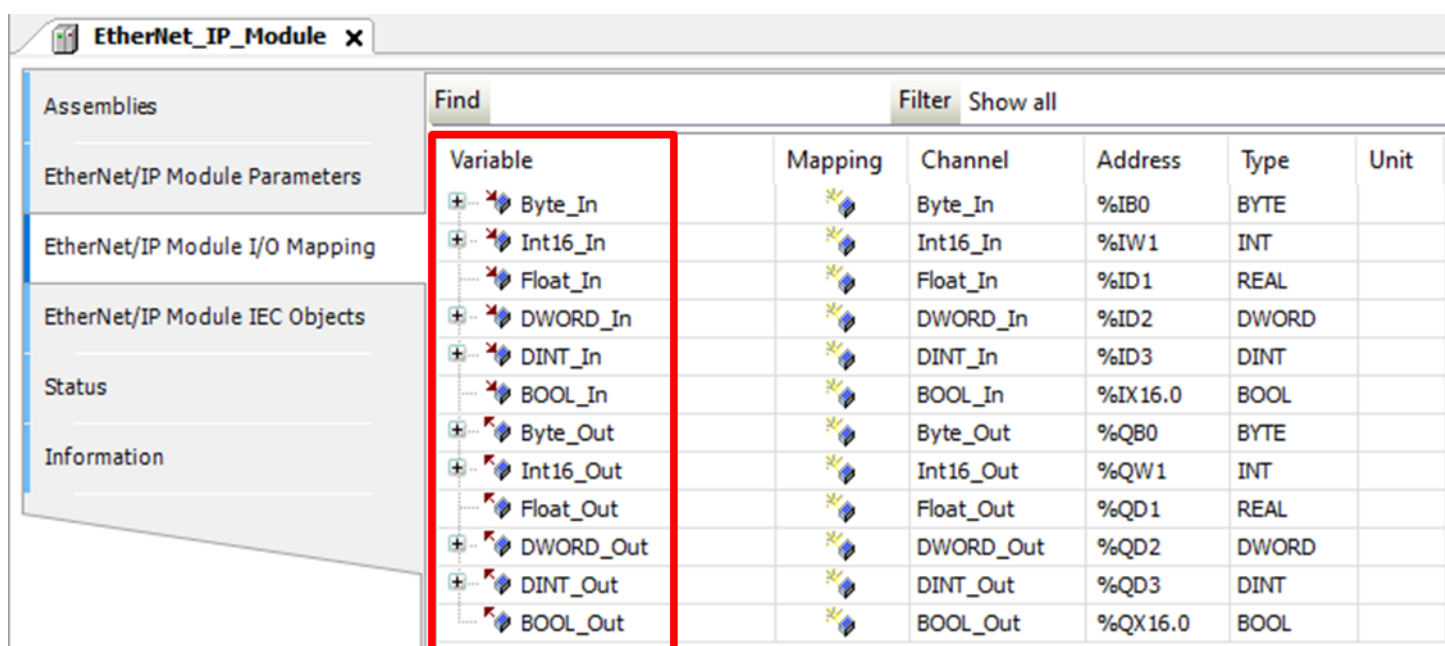


Общий список регистров можно увидеть в разделе **Ethernet/IP Module I/O Mapping**. Здесь же можно присвоить регистрам теги. Регистры типа %I** будут содержать данные от Мастера (Consuming Tags), а из регистров типа %Q** данные будут передаваться Мастеру (Producing Tags).



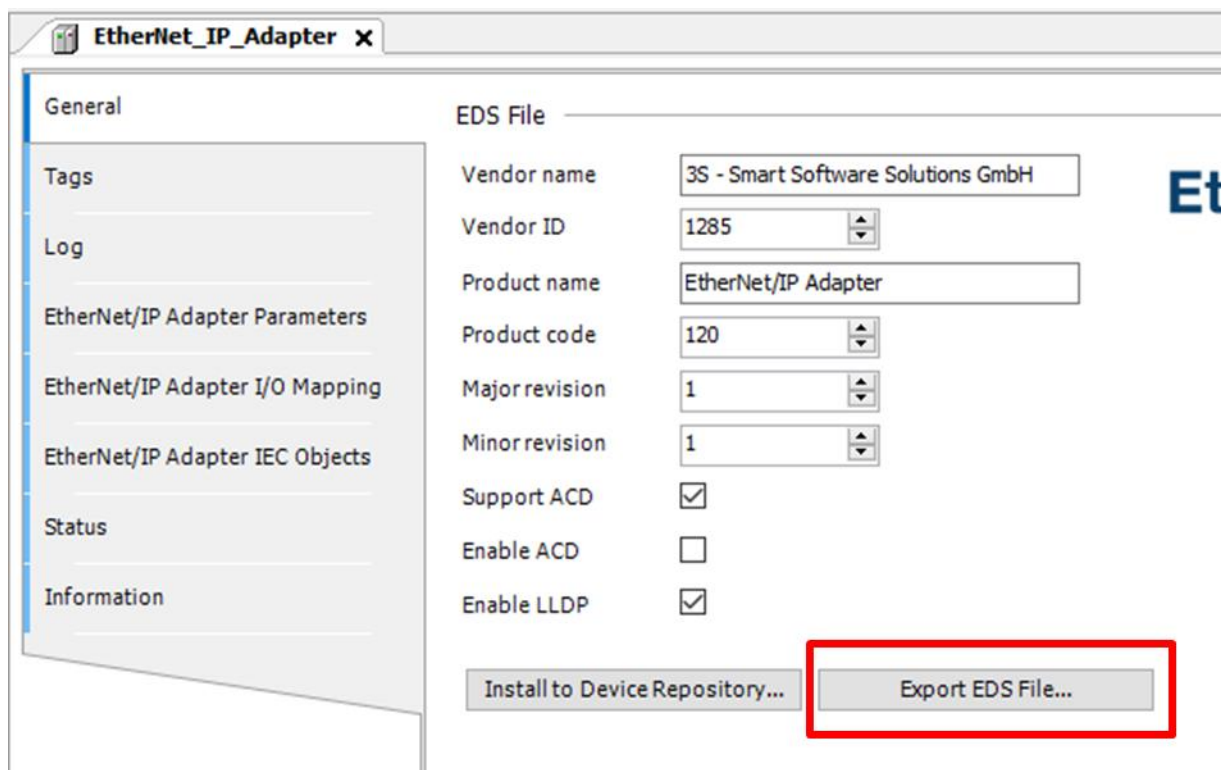
Variable	Mapping	Channel	Address	Type	Unit
Byte_In		Byte_In	%IB0	BYTE	
Int16_In		Int16_In	%IW1	INT	
Float_In		Float_In	%ID1	REAL	
DWORD_In		DWORD_In	%ID2	DWORD	
DINT_In		DINT_In	%ID3	DINT	
BOOL_In		BOOL_In	%IX16.0	BOOL	
Byte_Out		Byte_Out	%QB0	BYTE	
Int16_Out		Int16_Out	%QW1	INT	
Float_Out		Float_Out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

Для удобства лучше сразу присвоить тегам имена, которые можно будет использовать непосредственно в программе как Ведомого устройства (MH1000):



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



EtherNet_IP_Adapter X

General

Tags

Log

EtherNet/IP Adapter Parameters

EtherNet/IP Adapter I/O Mapping

EtherNet/IP Adapter IEC Objects

Status

Information

EDS File

Vendor name: 3S - Smart Software Solutions GmbH

Vendor ID: 1285

Product name: EtherNet/IP Adapter

Product code: 120

Major revision: 1

Minor revision: 1

Support ACD: ☒

Enable ACD: ☐

Enable LLDP: ☒

Install to Device Repository...

Export EDS File...

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 = "MH1048_EtherNet_IP Adapter";
```

И можно изменить имя файла:

 MH1048_EtherNet_IP Adapter.eds	24.07.2024 12:05	Файл "EDS"	25 КБ
--	------------------	------------	-------

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.

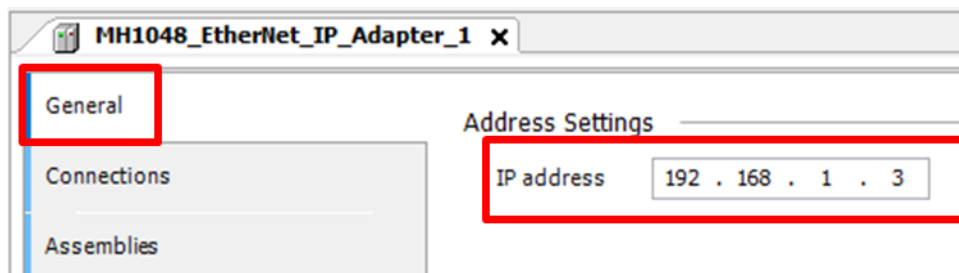
Импортируйте в репозиторий устройств полученный на предыдущем шаге XML файл с тегами от контроллера MH1048 (Adapter).

Name	Vendor
Fieldbuses	
EtherNet/IP	
EtherNet/IP Remote Adapter	
AH-CPU	
AS-CPU	
VFD	
AM400 Series PLC EIP Adapter	Inovance
AS00SCM-RTU(AS-FEN02)	Delta electronics, inc.
AX-308EA0MA1T	Delta electronics, inc.
CMM-EIP01/02	Delta electronics, inc.
Generic EtherNet/IP device	3S - Smart Software Solutions GmbH
MH1048_EtherNet_IP Adapter	3S - Smart Software Solutions GmbH

Добавьте устройство в проект:

Ethernet_1 (Ethernet)
EtherNet_IP_Scanner_1 (EtherNet/IP Scanner)
MH1048_EtherNet_IP_Adapter_1 (MH1048_EtherNet_IP Adapter)

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 МН1048). Адреса можно посмотреть в пункте **EtherNet/IP I/O Mapping**:

General

Connections

Assemblies

User-Defined Parameters

Log

EtherNet/IP Parameters

EtherNet/IP I/O Mapping

EtherNet/IP IEC Objects

Status

Find

Filter

Show all

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

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:

General	Find	Filter	Show all		
Connections	Variable	Mapping	Channel	Address	Type
Assemblies	Connection1				
User-Defined Parameters	Byte_In		Byte_In	%IB268	BYTE
Log	Int16_In		Int16_In	%IW135	INT
EtherNet/IP Parameters	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
EtherNet/IP I/O Mapping	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
EtherNet/IP IEC Objects					
Status					

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

Watch 1			
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**

MH1048_EtherNet_IP_Adapter_1.eState

Данные автоматически будут попадать в регистры Ведомого устройства MH1048 (Consumed Tags):

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

В программе Ведомого состояния связи с Мастером можно также контролировать через элемент структуры **Device_Name.eState**:

EtherNet_IP_Module.eState

И наоборот, если в программе Ведомого (МН1048) задать значения для исходящих тегов (Produced Tags):

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

то данные автоматически попадут в регистры Мастера (Delta AX-308E):

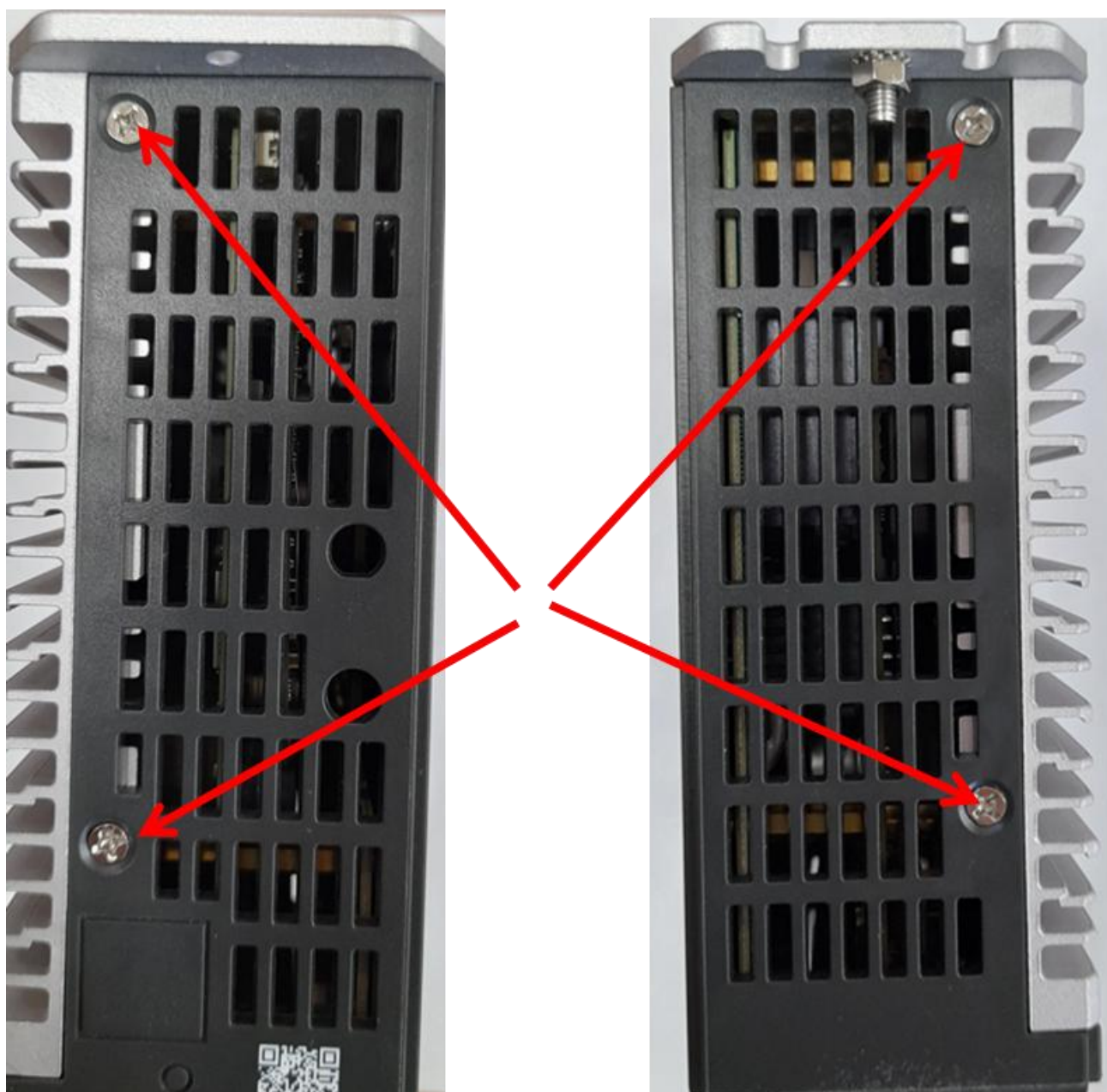
Watch 1			
Expression	Application	Type	Value
 MX308_EtherNet_IP_Adapter.eState	Device.Application	ADAPTERSTATE	RUNNING
 IoConfig_Globals_Mapping.Byte_Out	Device.Application	BYTE	0
 IoConfig_Globals_Mapping.Int16_Out	Device.Application	INT	0
 IoConfig_Globals_Mapping.Float_Out	Device.Application	REAL	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

Замена батарейки часов реального времени

В контроллерах типа МН1048 используется стандартная батарейка CR2032. Идёт в комплекте с контроллером. Срок службы примерно 3 года. Используется только для работы часов реального времени (для поддержания памяти не требуется). Если Вы не используете часы реального времени в контроллере, то наличие батарейки не требуется.

Отсек с батарейкой находится внутри корпуса контроллера на плате в держателе (батхолдере). Для извлечения батарейки необходимо выполнить несколько шагов:

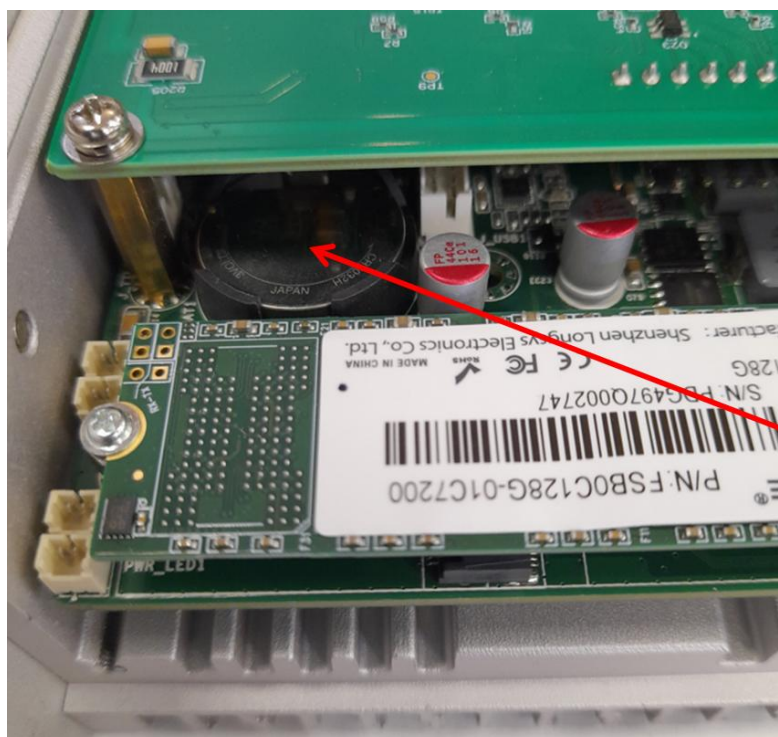
Шаг 1. Открутите 4 винта на корпусе контроллера, по 2 винта на каждой стороне (верхней и нижней):



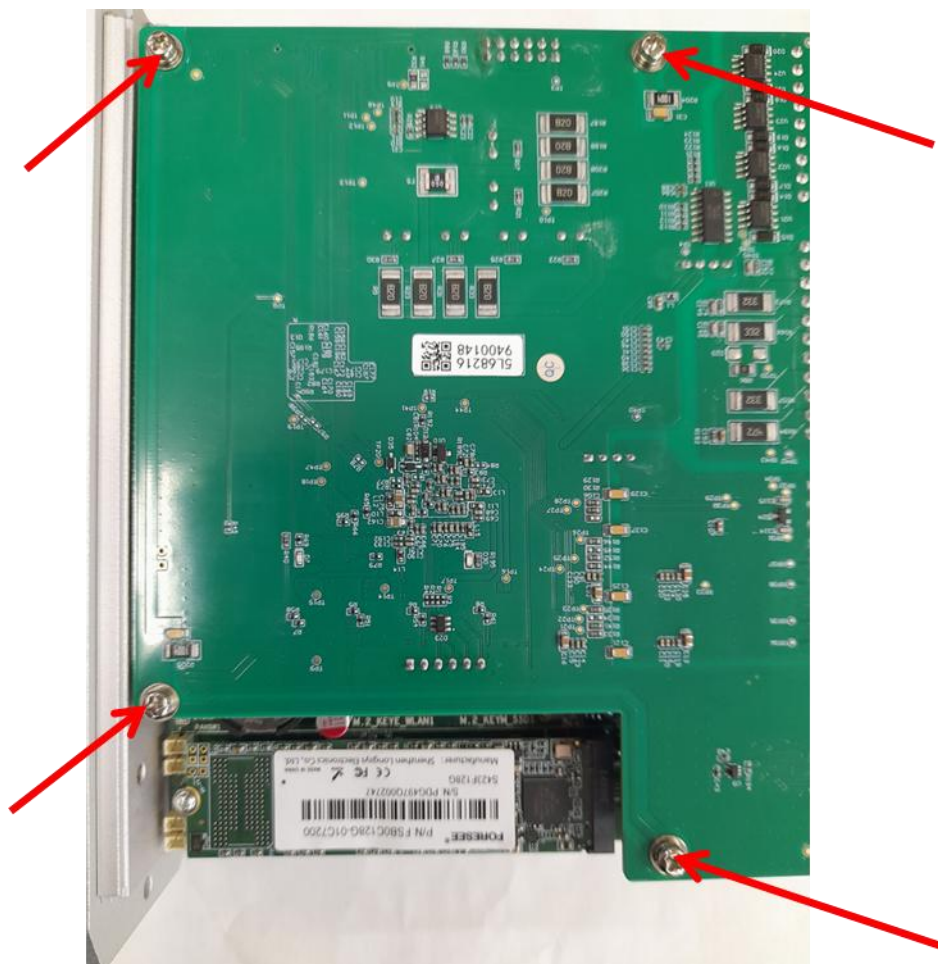
Шаг 2. Сдвинуть внешнюю оболочку корпуса вниз:



Батарейка находится в держателе между двумя платами.

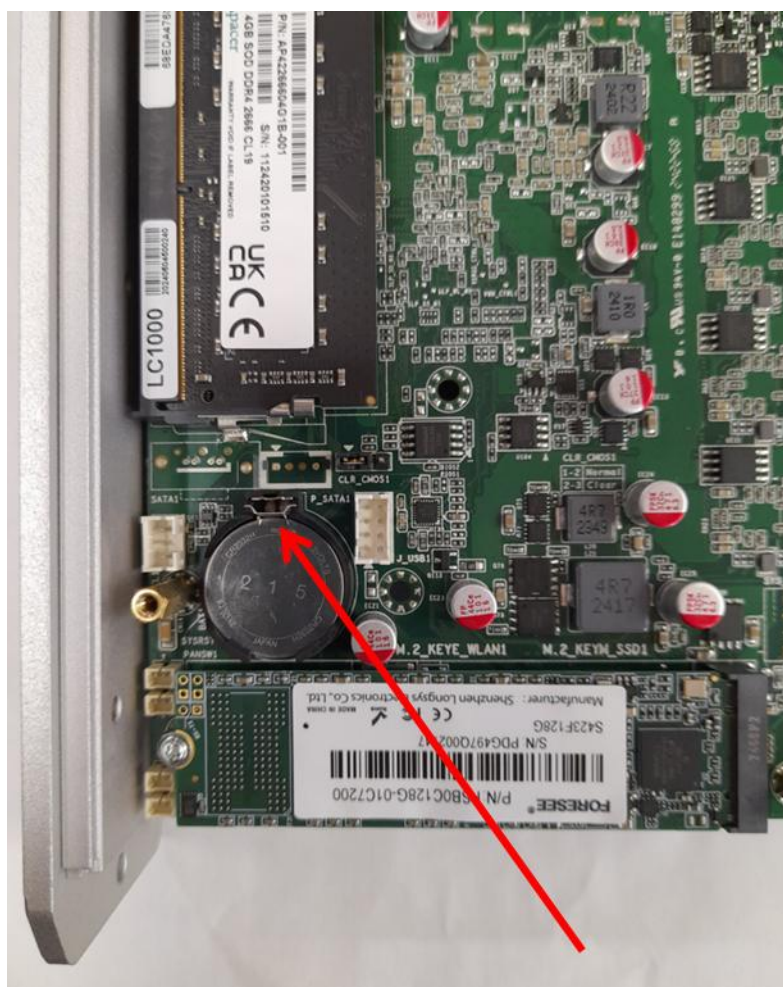


Шаг 3. Открутите 4 винта и снимите верхнюю плату:



Внимание! Не пытайтесь извлечь батарейку не снимая верхней платы!!! Это может привести к повреждению контроллера!

Шаг 4. Отогните фиксатор и извлеките батарейку.



Вставьте в держатель новую батарейку. Сборка осуществляется в обратном порядке.

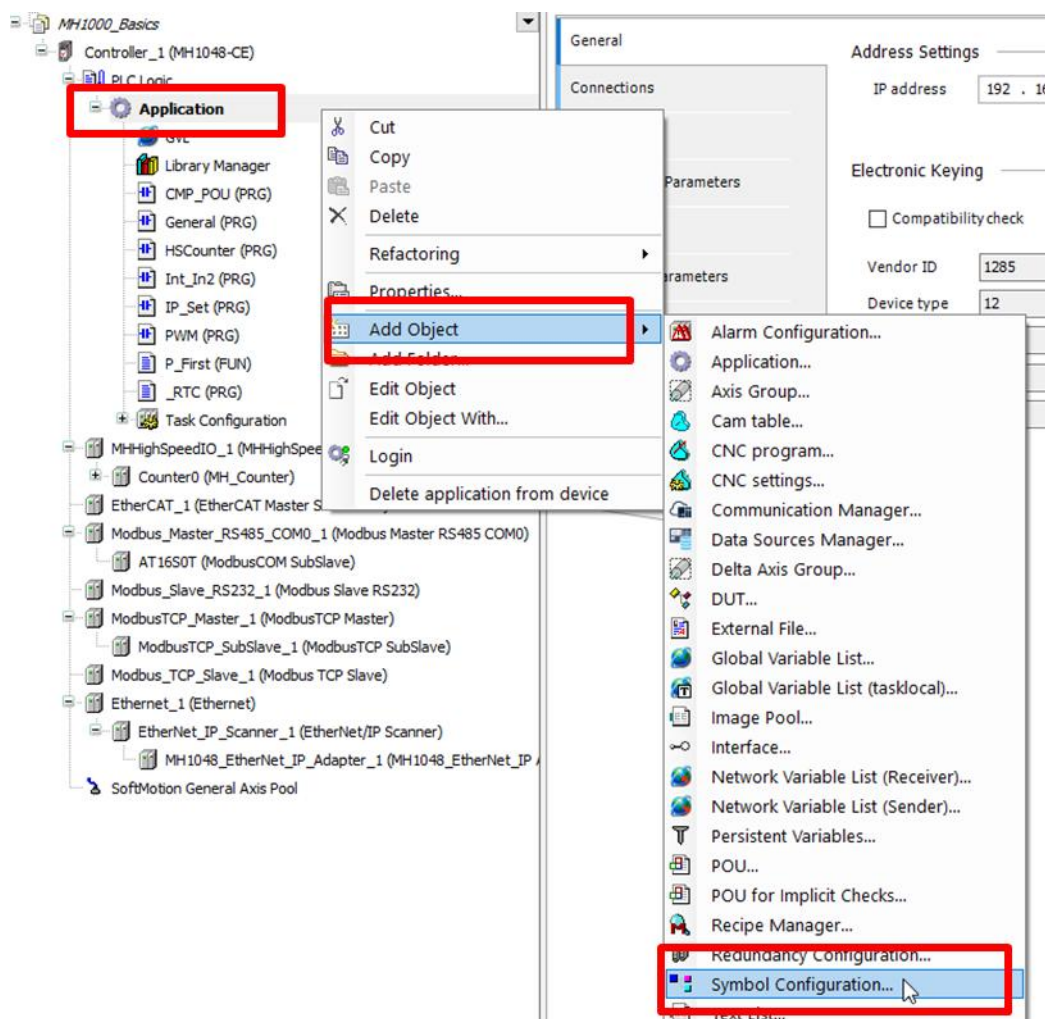
Обмен тегами CODESYS с панелью оператора

В среде программирования CODESYS существует стандартная процедура экспорта тегов в виде XML файла для дальнейшего использования в панели оператора или SCADA, которые поддерживают данный протокол связи с устройствами CODESYS. Панель (SCADA) будет выступать в роли Мастера (Клиента), а контроллер в роли Ведомого (Сервера).

Откройте любой проект для контроллера MH1048, в котором есть созданные переменные.

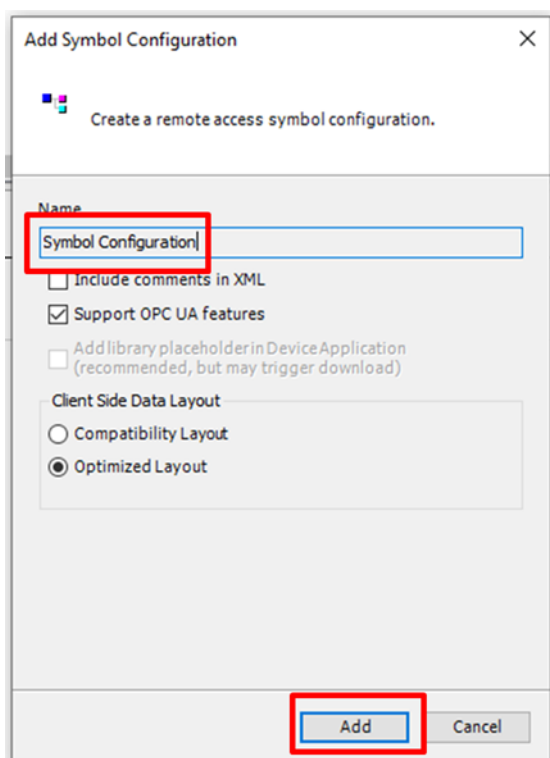
	Scope	Name	Address	Data type
2	VAR	Coil1		BOOL
3	VAR	Int1		INT
4	VAR	Int2		INT
5	VAR	rARRAY		ARRAY[0..2999] OF REAL
6	VAR	Byte1		BYTE
7	VAR	Byte2		BYTE
8	VAR	str1		string(20)
9	VAR	wstr		wstring(20)

Щёлкните правой кнопкой мышки на вкладки **Application** и в открывшемся меню выберите пункт **Add Object** и в следующем меню пункт **Symbol Configuration**.

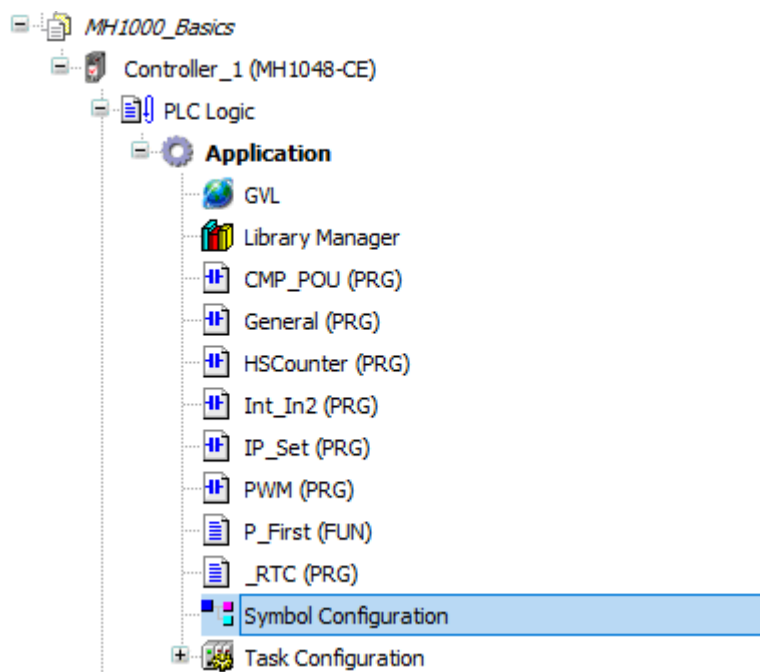


Внимание: В составе одного приложения (Application) может быть только одна таблица символов.

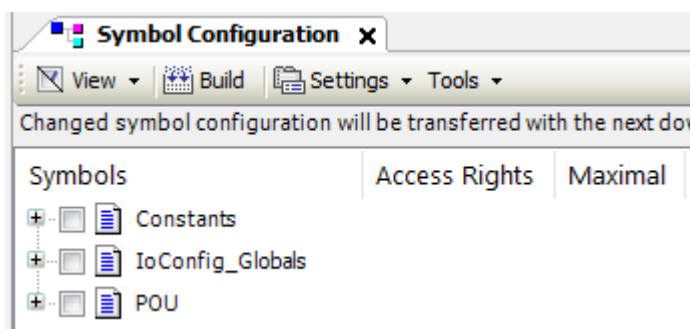
В появившемся окне определите название и нажмите **Add**:



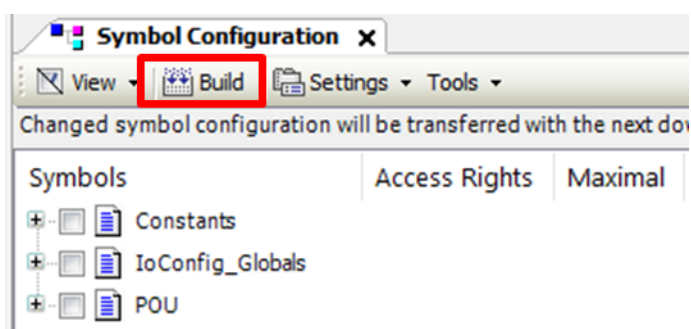
В разделе Application будет создан пункт с этим названием (в нашем примере используется название по умолчанию **Symbol Configuration**):



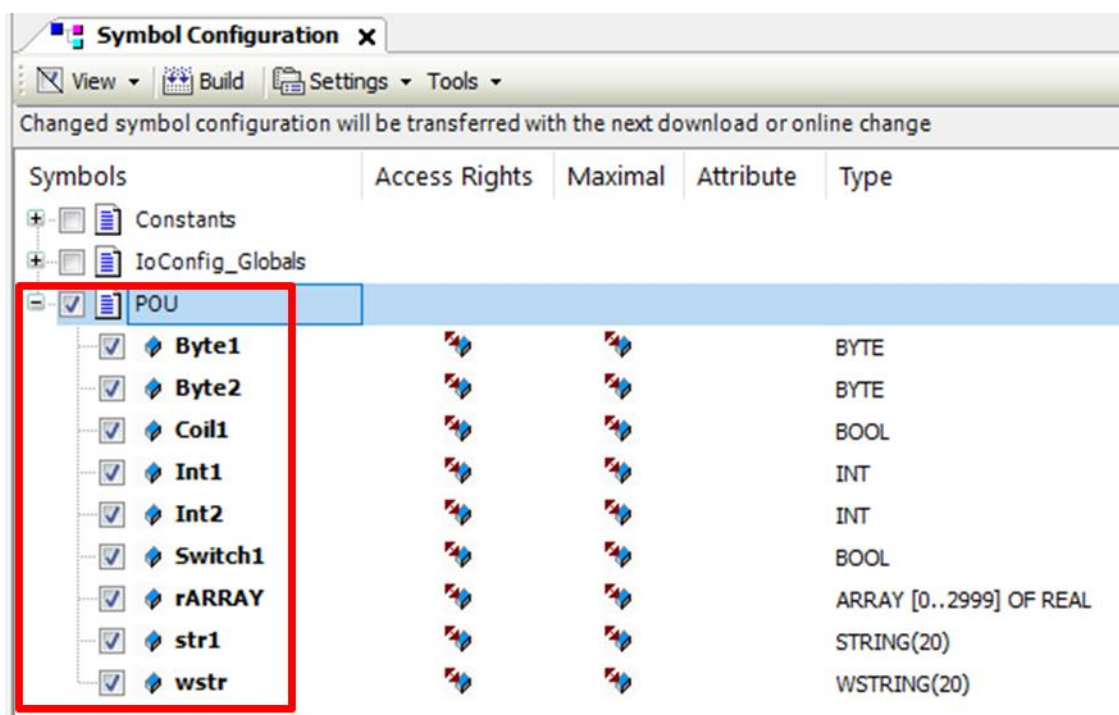
Также система сразу откроет одноименную вкладку:



Нажмите кнопку **Build** для построения проекта и формирования общего списка переменных:



После компиляции отметьте флажками необходимые переменные:



Нажмите ещё раз кнопку **Build**.

В папке проекта будет создан XML файл вида:

MH1000_CODESYS_TAGss.Controller_1.Application.xml



	MH1000_Basics.cleaned	16.05.2025 14:59
	MH1000_Basics.Controller_1.Application.xml	16.05.2025 14:59
	MH1000_Basics.Controller_1.Sim.Controller_1.Applicati...	16.04.2025 13:49
	MH1000_Basics.Controller_1.Sim.Controller_1.Applicati...	16.04.2025 13:49
	MH1000_Basics.project	16.05.2025 14:59

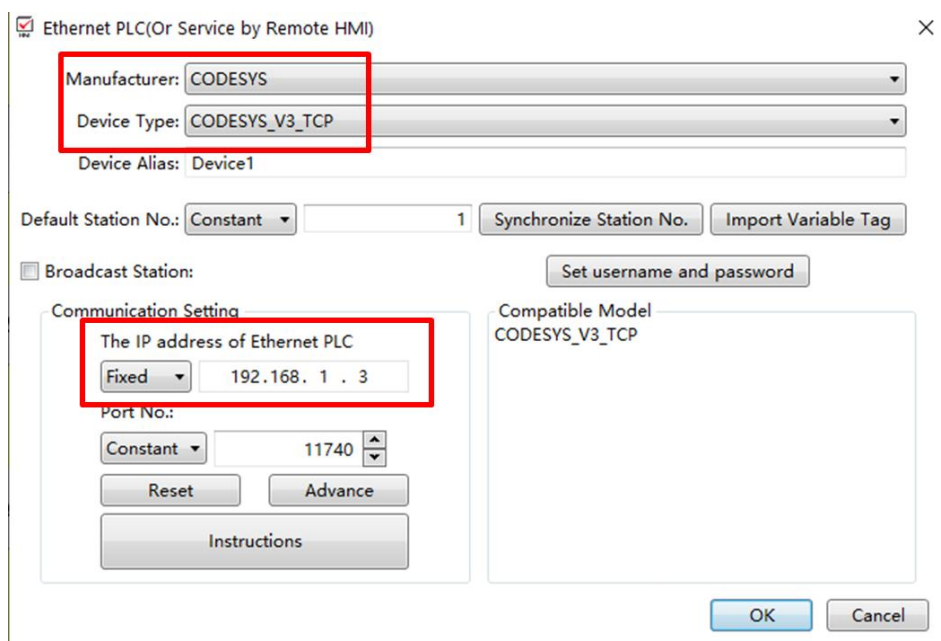
В созданном XML файле будут содержаться выбранные переменные:

```
<NodeList>
  <Node name="Application">
    <Node name="POU">
      <Node name="Byte1" type="T_BYTE" access="ReadWrite" />
      <Node name="Byte2" type="T_BYTE" access="ReadWrite" />
      <Node name="Coil1" type="T_BOOL" access="ReadWrite" />
      <Node name="Int1" type="T_INT" access="ReadWrite" />
      <Node name="Int2" type="T_INT" access="ReadWrite" />
      <Node name="rARRAY" type="T_ARRAY__0_2999_OF_REAL" access="ReadWrite" />
      <Node name="str1" type="T_STRING_20_" access="ReadWrite" />
      <Node name="Switch1" type="T_BOOL" access="ReadWrite" />
      <Node name="wstr" type="T_WSTRING_20_" access="ReadWrite" />
    </Node>
  </Node>
</NodeList>
</Symbolconfiguration>
```

Загрузите проект в контроллер, чтобы таблица с тегами стала доступна для Мастера (Клиента)!

Данный XML файл можно импортировать в среду разработки проекта панели оператора или SCADA, которые поддерживают технологию связи через теги CODESYS. Для примера рассмотрим панель оператора Optimus Drive OTP210-070E, среда разработки Optimus Drive HMI Soft 3.

Создайте проект для панели OTP210-070E и в разделе настройки связи выберите Ethernet PLC и драйвер CODESYS. Задайте IP адрес (в нашем примере 192.168.1.3):



Нажмите кнопку **Import Variable Tag** и импортируйте XML файл с тегами:

No. **Import Variable Tag**

me and password

В открывшемся окне нажмите кнопку **Browse**:

☒ Import Tag

Select Tag

File Name: **Browse**

выберите XML файл:

My Documents > OptimusDrive > MH1000 > MH1000_Basics

Новая папка

Имя	Дата изменения
MH1000_Basics.Controller_1.Application.xml	16.05.2025 14:59

Разверните список тегов и поставьте флажки, затем нажмите кнопку **Import**:

☒ Import Tag

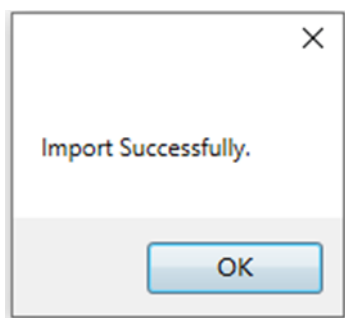
Select Tag

File Name: D:\My Documents\OptimusDrive\MH1000\MH1000_Basics\MH1000_ **Browse** **Import**

Device_tags

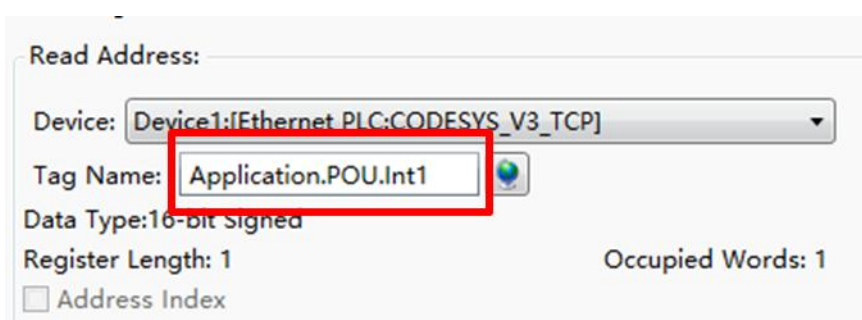
☒ POU

При удачном импорте тегов появится окошко с надписью **Import Successfully**:

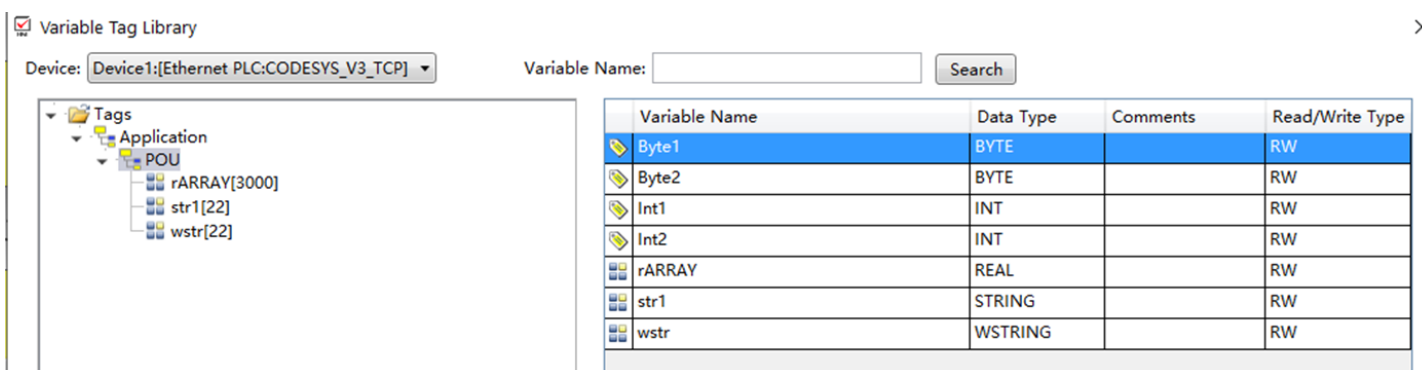


Нажмите **OK**.

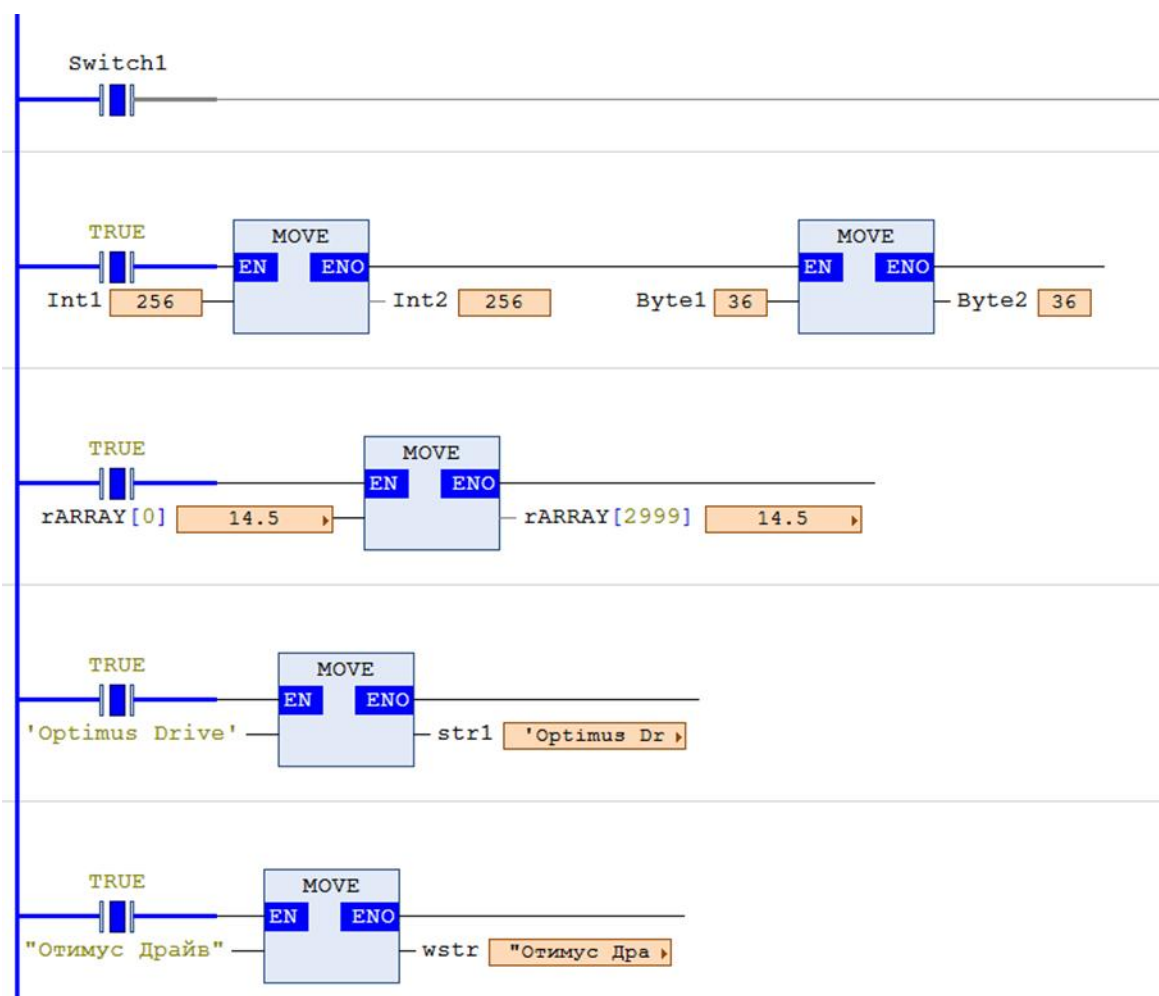
Теперь теги будут доступны как адреса у экранных объектах:



Создайте необходимое количество экранных объектов, указывая в качестве адресов импортированные теги:



Загрузите в панель или запустите онлайн симулятор. На экране будут отображаться те же данные, что и в программе контроллера.

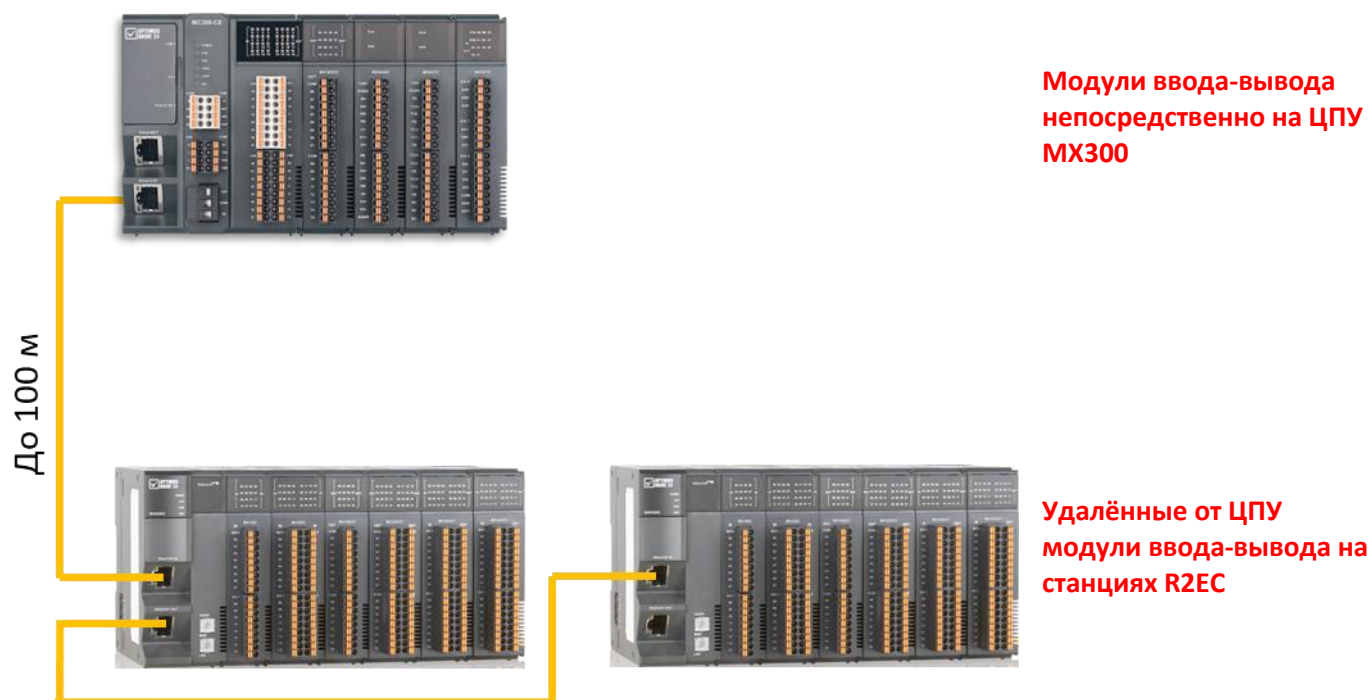


Работа со станцией удалённого ввода-вывода R2EC в сети EtherCAT

Станция (каплер) **R2EC** позволяет удалённо использовать модули ввода-вывода от контроллера MX300. Физически модули подсоединяются к станции также как к ЦПУ. Далее станция соединяется с ЦПУ кабелем EtherCAT.

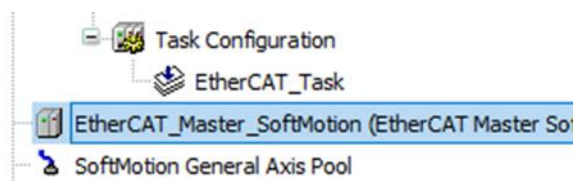
В проект контроллера добавляется устройства типа R2EC и уже на него добавляются модули ввода-вывода. Обращение к модулям идёт через станцию, т.е. сетевой адрес есть только у станции. Модули находятся на её внутренней шине и не имеют прямого выхода в сеть.

Технология удалённых станций позволяет использовать дискретные и аналоговые сигналы с вынесенных от ЦПУ модулей ввода-вывода как если бы они были непосредственно на ЦПУ. Т.е. сразу через переменные в программе, без коммуникационных запросов.

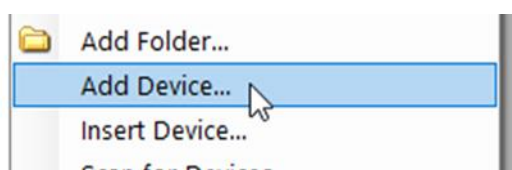


Рассмотрим порядок работы с удалёнными станциями. Сначала создайте проект и добавьте в него устройство типа ЦПУ MX300 и адаптер EtherCAT, как описано в соответствующих Разделах данного Руководства.

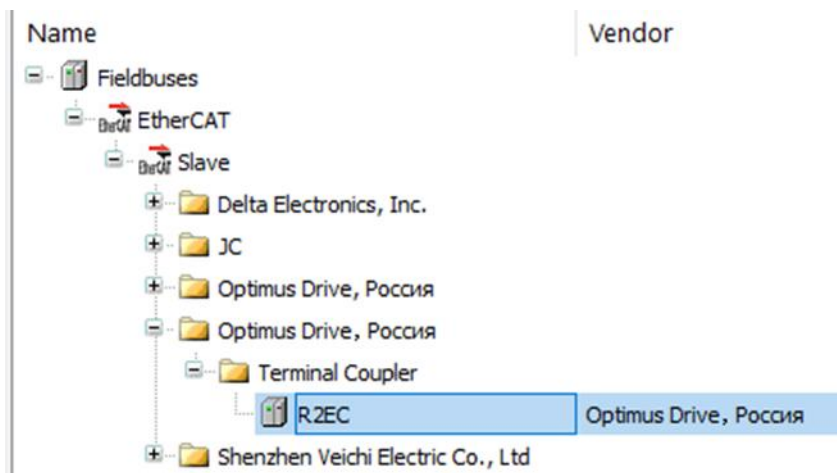
Далее выберите в древе проекта пункт EtherCAT Master и правой кнопкой мышки вызовите меню :



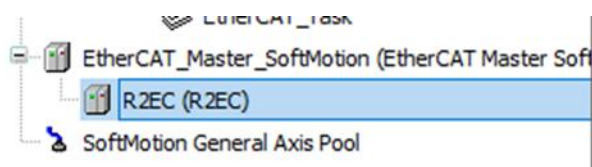
В открывшемся окне выберите пункт :



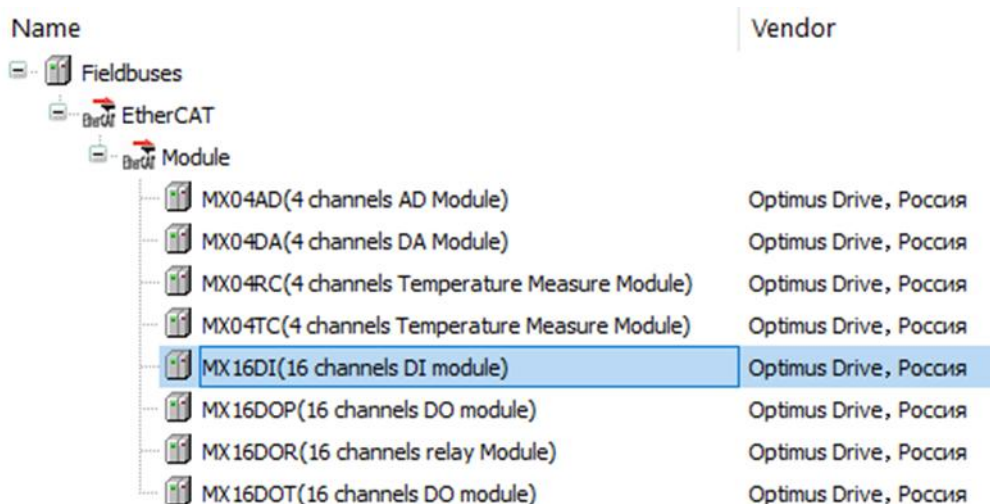
В открывшемся окне выберите **Fieldbuses - EtherCAT Slave – Optimus Drive – Terminal Coupler - R2EC**:



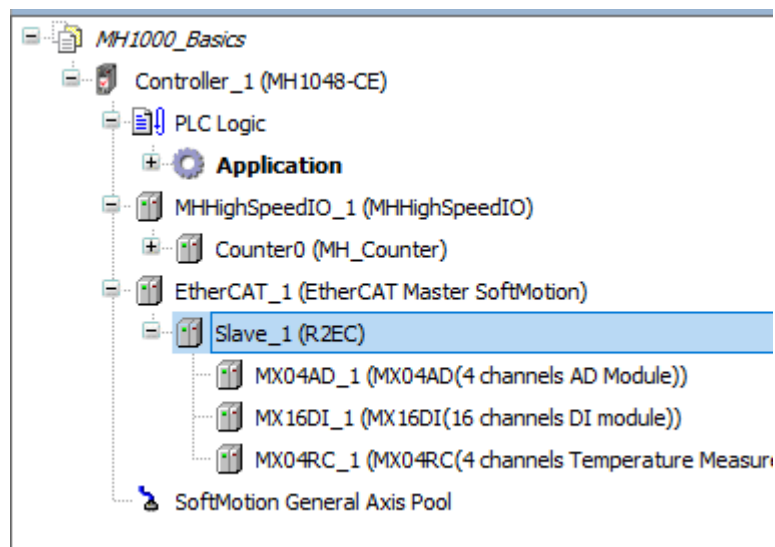
В древе проекта появится пункт:



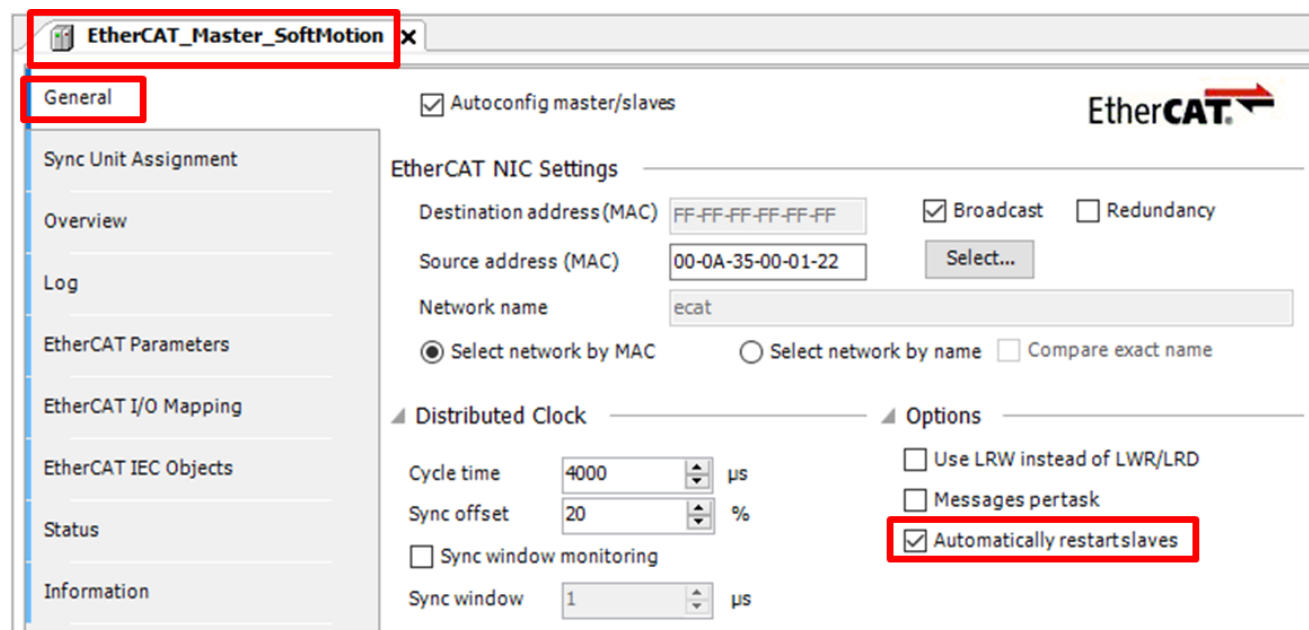
На этом этапе мы добавили станцию удалённого ввода-вывода. Далее не неё необходимо добавить нужные модули расширения. Для этого щёлкните правой кнопкой мышки на пункте **R2EC** и в открывшемся окне выберите нужные модули:



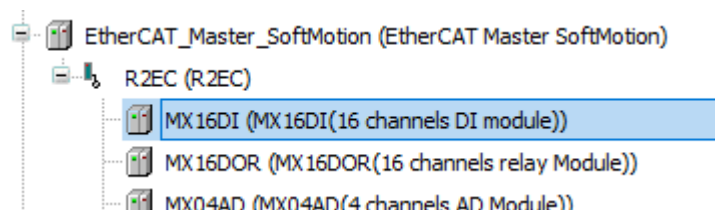
В итоге в древе проекта сформируется нужный состав оборудования, например такой:



Для автоматического подхвата станции необходимо поставить флажок:



Параметры любого из модулей можно посмотреть и настроить открыв страницу данного модуля. Для этого щёлкните дважды левой кнопкой мышки на названии модуля в древе проекта, например на MX16DI:



Откроется вкладка с параметрами модуля.

MX16DI X

Module I/O Mapping

Module IEC Objects

Information

General

Name: MX16DI(16 channels DI module)
Vendor: Optimus Drive, Россия
Categories: Module
Type: 68
ID: 4321_61100025_0_61400005
Version: 0
Order number: MX16DI
Description: EtherCAT Module imported from Slave

Параметры любого модуля включают вкладку **Module I/O Mapping**, в которой можно посмотреть регистры, которые выделила система для чтения/записи данных с модуля, а также привязать к ним свои переменные. В программе контроллера пользоваться предпочтительно переменными, а не физическими регистрами. Например, для модуля MX16DI данная вкладка выглядит так:

MX16DI X

Module I/O Mapping

Module IEC Objects

Information

Find

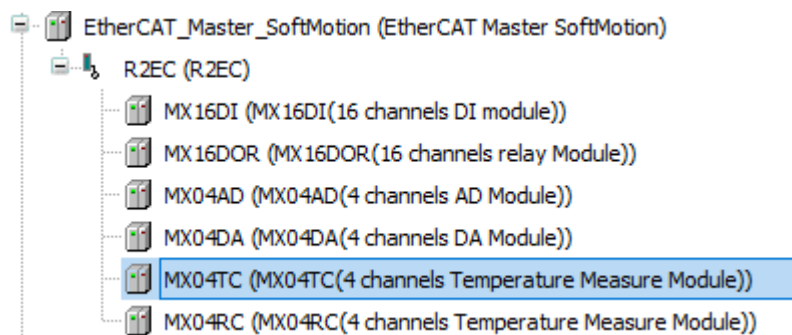
Filter

Show all

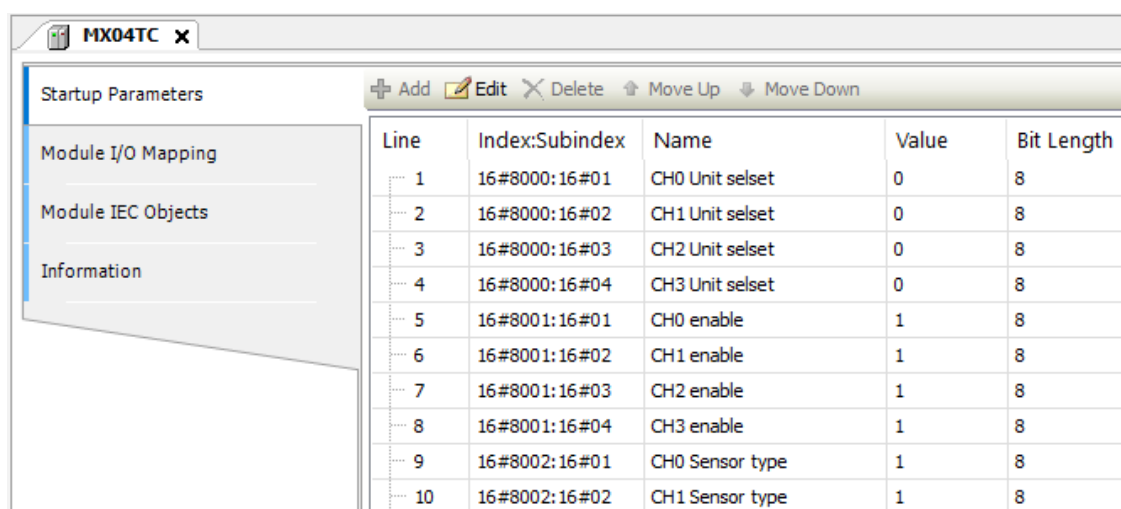
Variable	Mapping	Channel	Address	Type
16#1A00 input mapping				
DI_1_0		IN BIT 1	%IX4.0	BIT
DI_1_1		IN BIT 2	%IX4.1	BIT
		IN BIT 3	%IX4.2	BIT
		IN BIT 4	%IX4.3	BIT
		IN BIT 5	%IX4.4	BIT
		IN BIT 6	%IX4.5	BIT
		IN BIT 7	%IX4.6	BIT
		IN BIT 8	%IX4.7	BIT
		IN BIT 9	%IX5.0	BIT
		IN BIT 10	%IX5.1	BIT
		IN BIT 11	%IX5.2	BIT
		IN BIT 12	%IX5.3	BIT
		IN BIT 13	%IX5.4	BIT
		IN BIT 14	%IX5.5	BIT
		IN BIT 15	%IX5.6	BIT
		IN BIT 16	%IX5.7	BIT

В колонке **Address** указаны физические регистры, которые система привязала к входам, а в колонке **Variable** можно дать свои названия переменным, через которые можно будет обращаться к входам в программе.

В аналоговых и температурных модулях появляется ещё вкладка **Startup Parameters** с настройками рабочих режимов. Например, откройте двойным щелчком левой кнопки мышки термодатчикный модуль MX04TC:

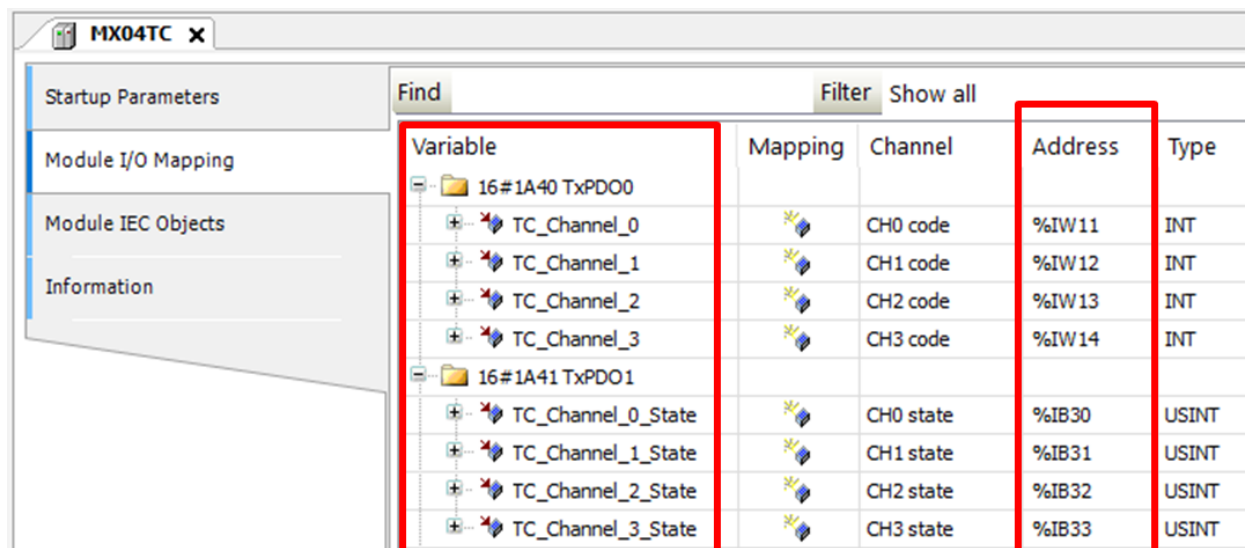


Откроется вкладка с настройками модуля:



В пункте **Startup Parameters** можно задать рабочий режим каждого измерительного канала такие как тип датчика, единицы измерения и т.п. Параметры приведены в настоящем Руководстве в Главе с описанием температурных модулей.

В пункте **Module I/O Mapping** можно посмотреть регистры, которые выделила система для чтения/записи данных с модуля, а также привязать к ним свои переменные:

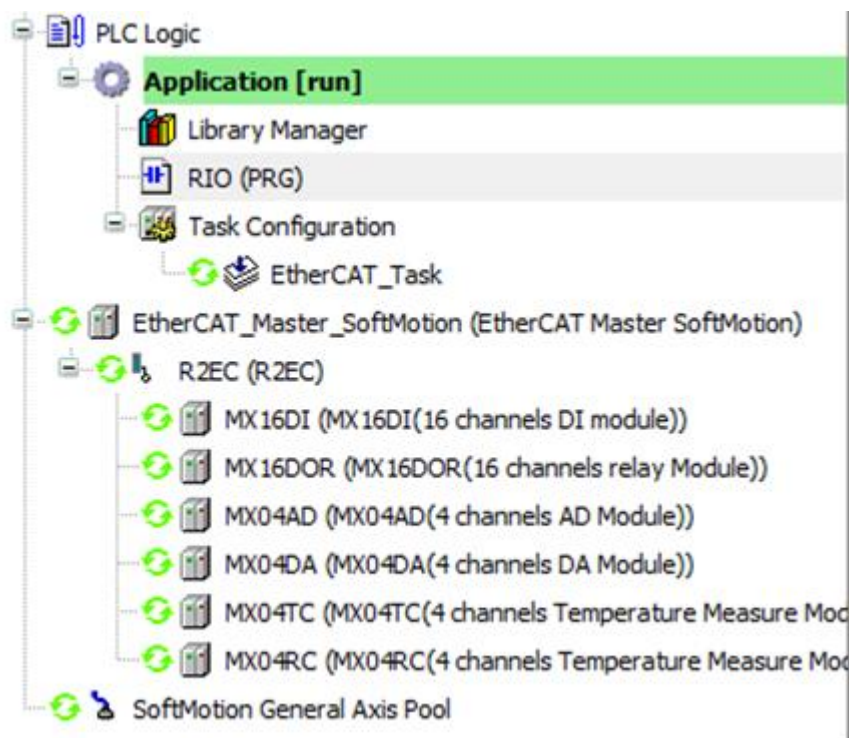


Аналогичным образом строится работы со всеми модулями.

Далее, установите модули на станцию строго в той последовательности как они сконфигурированы в проекте. Соедините ЦПУ и станцию EtherCAT кабелем. Подайте на ЦПУ, станцию и аналоговые/температурные модули питание 24 VDC.

Установите соединение с ЦПУ и загрузите проект с нужной конфигурацией станции. Войдите в онлайн режим с контроллером (см. настоящее Руководство).

Если все настройки были сделаны правильно, то все устройства должны быть подсвечены зелёным цветом:



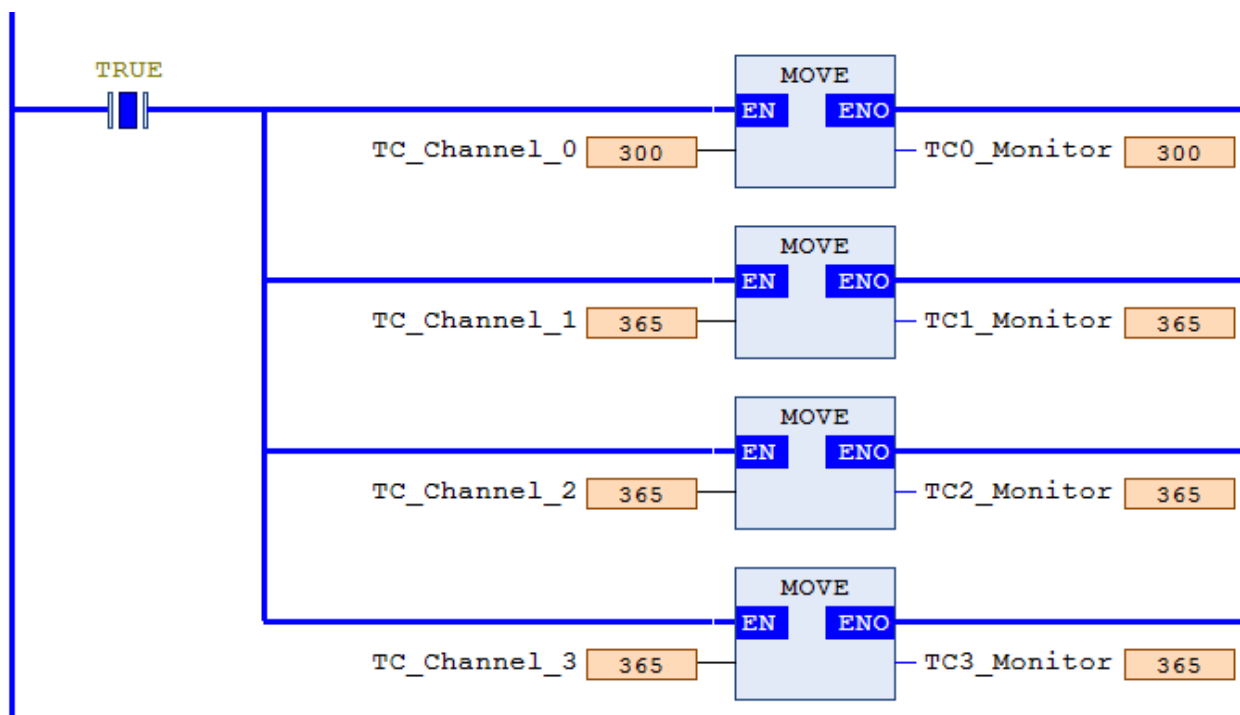
В онлайн режиме будет доступна колонка с текущим состоянием каналов ввода-вывода, например MX16DI:

MX16DI x						
Module I/O Mapping						
Find Filter Show all Add FB for I						
Module IEC Objects	Variable	Mapping	Channel	Address	Type	Current Value
Information	16#1A00 input mapping					
	DI_1_0		IN BIT1	%IX4.0	BIT	FALSE
	DI_1_1		IN BIT2	%IX4.1	BIT	FALSE
			IN BIT3	%IX4.2	BIT	FALSE
			IN BIT4	%IX4.3	BIT	FALSE
			IN BIT5	%IX4.4	BIT	FALSE
			IN BIT6	%IX4.5	BIT	FALSE
			IN BIT7	%IX4.6	BIT	FALSE
			IN BIT8	%IX4.7	BIT	FALSE
			IN BIT9	%IX5.0	BIT	FALSE
			IN BIT10	%IX5.1	BIT	FALSE
			IN BIT11	%IX5.2	BIT	FALSE
			IN BIT12	%IX5.3	BIT	FALSE
			IN BIT13	%IX5.4	BIT	FALSE
			IN BIT14	%IX5.5	BIT	FALSE
			IN BIT15	%IX5.6	BIT	FALSE
			IN BIT16	%IX5.7	BIT	FALSE

Текущие значения на измерительных каналах модуля термодатчиков MX04TC:

MX04TC x						
Startup Parameters	Find	Filter	Show all	+ Add FB fo		
Module I/O Mapping	Variable	Mapping	Channel	Address	Type	Current Value
Module IEC Objects	16#1A40 TxPDO0					
Information	TC_Channel_0		CH0 code	%IW11	INT	307
	TC_Channel_1		CH1 code	%IW12	INT	374
	TC_Channel_2		CH2 code	%IW13	INT	375
	TC_Channel_3		CH3 code	%IW14	INT	374
	16#1A41 TxPDO1					
	TC_Channel_0_State		CH0 state	%IB30	USINT	0
	TC_Channel_1_State		CH1 state	%IB31	USINT	0
	TC_Channel_2_State		CH2 state	%IB32	USINT	0
	TC_Channel_3_State		CH3 state	%IB33	USINT	0

В программе контроллера данные можно использовать через созданные переменные:

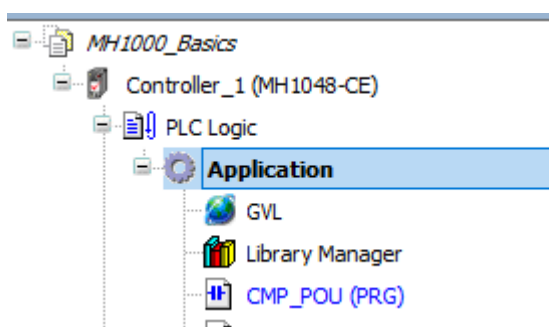


Связь по протоколу OPC UA в режиме сервера

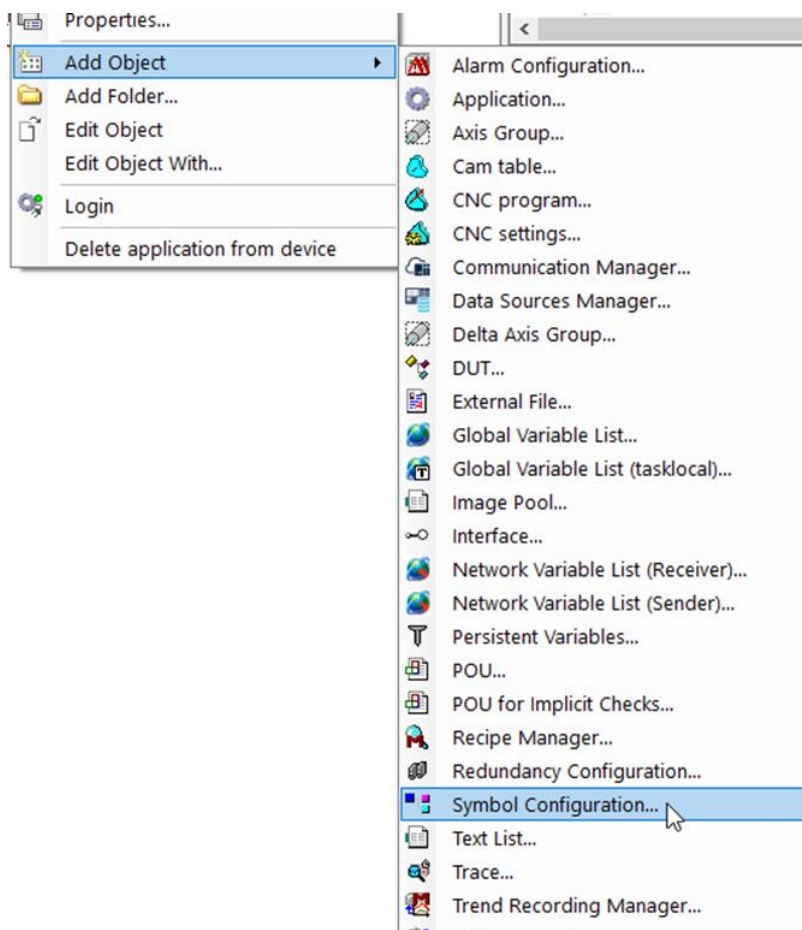
Протокол **OPC UA** (Unified Architecture) – это современный широко распространённый стандарт, описывающий передачу данных в промышленных сетях, анонсированный в 2008 году организацией OPC Foundation и закреплённый в стандарте IEC62541. Является кросс-платформенным протоколом клиент-серверной архитектуры. Контроллеры серии МН1048 поддерживают **OPC UA Server** (Ведомый, отвечает на запросы Мастера).

Для начала работы создайте проект, добавьте устройство (контроллер МН1048) и адаптер Ethernet, создайте POU, добавьте несколько переменных, установите связь с контроллером и загрузите проект (см. соответствующие разделы настоящего Руководства).

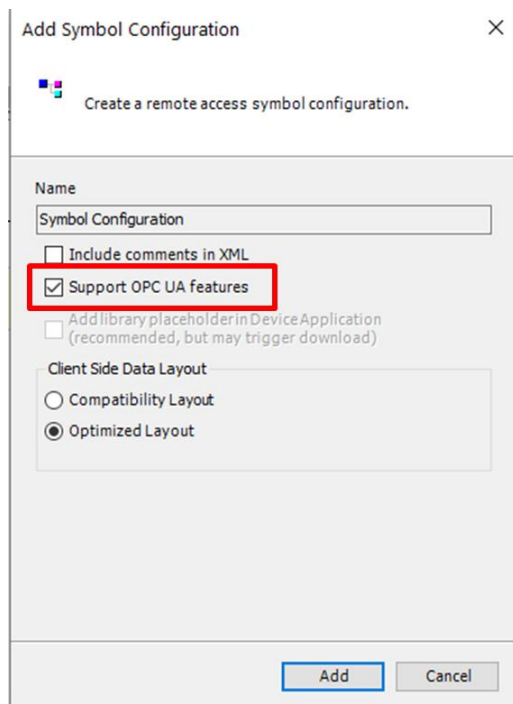
В древе проекта выберите пункт **Application** и нажмите правую кнопку мышки:



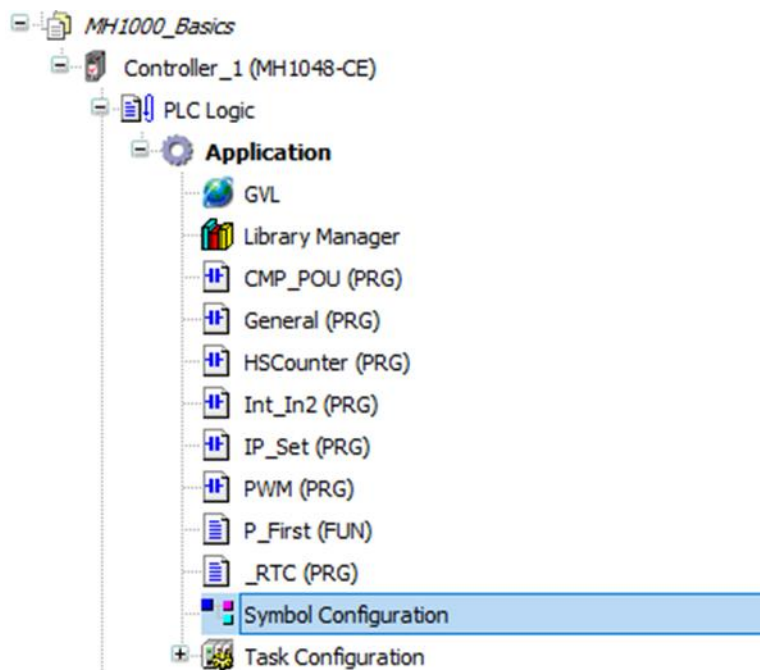
В появившемся меню выберите пункт **Add Object – Symbol Configuration**:



В открывшемся окне поставьте флажок **Support OPC UA features** и нажмите кнопку **Add**:



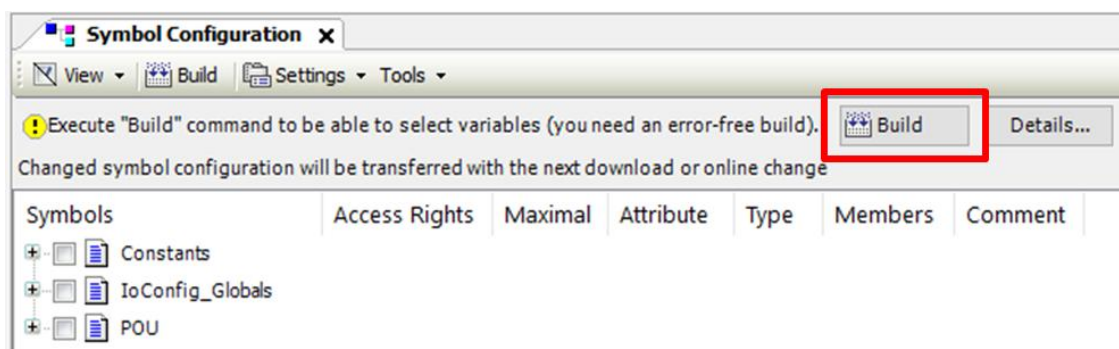
В древе проекта появится пункт **Symbol Configuration**:



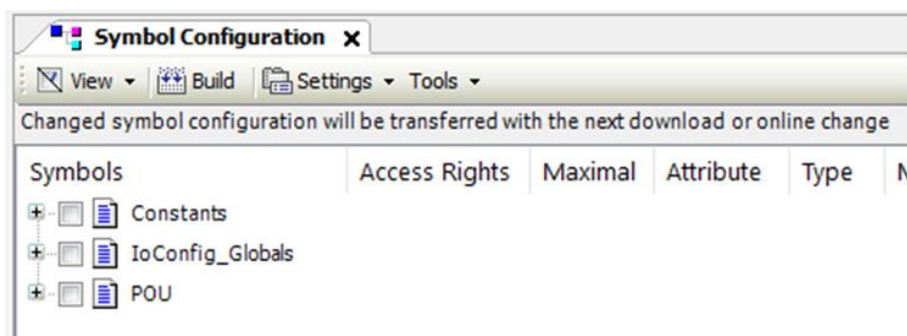
Для наглядности в нашем проекте созданы переменные различных типов данных:

Scope	Name	Address	Data type
VAR	bool_coil		BOOL
VAR	bool_contact		BOOL
VAR	byte_IN		BYTE
VAR	byte_OUT		BYTE
VAR	dint_IN		dint
VAR	dint_OUT		dint
VAR	lint_IN		LINT
VAR	lint_OUT		LINT
VAR	lreal_IN		LREAL
VAR	lreal_OUT		LREAL
VAR	real_IN		REAL
VAR	real_OUT		REAL
VAR	string1		string(20)
VAR	string2		string(20)
VAR	word_IN		word
VAR	word_OUT		word
VAR	wstring1		wstring(20)
VAR	wstring2		wstring(20)

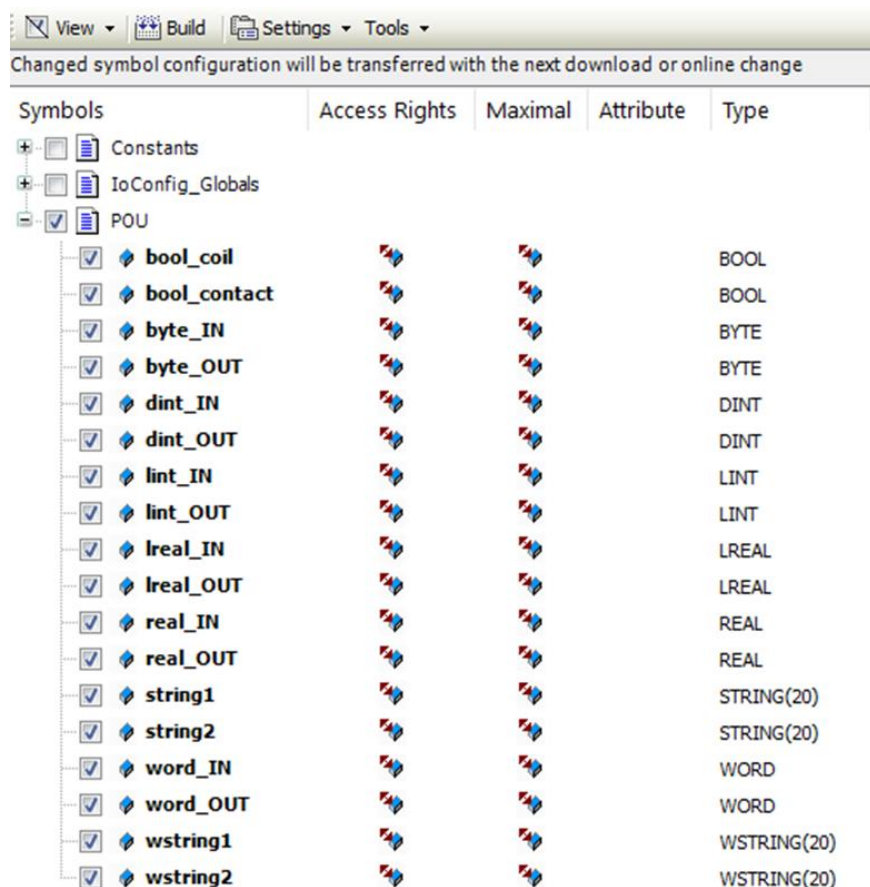
Для того, чтобы сделать данные переменные доступными по протоколу OPC UA Server необходимо двойным щелчком левой кнопки мышки открыть пункт **Symbol Configuration**. В открывшейся вкладке необходимо выполнить компиляцию проекта путём нажатия кнопки **Build**. Проект соответственно должен быть готов к компиляции. Без неё переменные не станут доступны для чтения-записи.



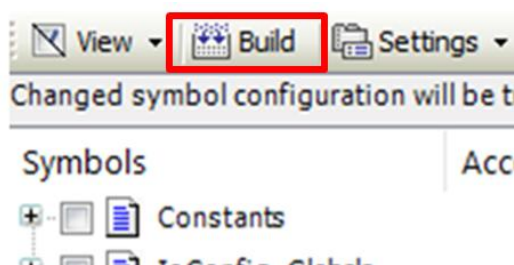
После удачной компиляции вкладка примет такой вид:



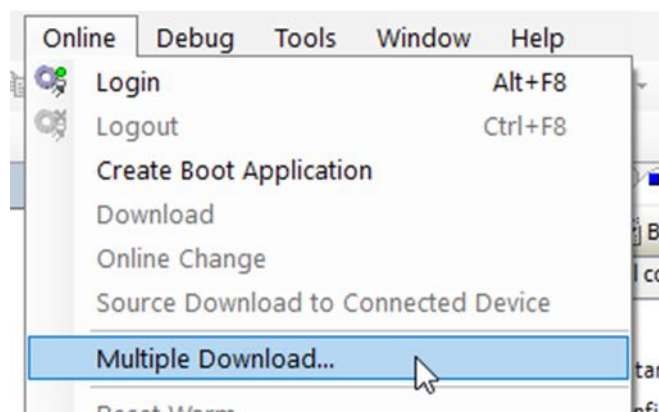
После компиляции необходимо развернуть список и отметить нужные переменные, в нашем примере все. Тип доступа (**Access Rights**) Read-Write.



После этого необходимо снова выполнить компиляцию проекта:

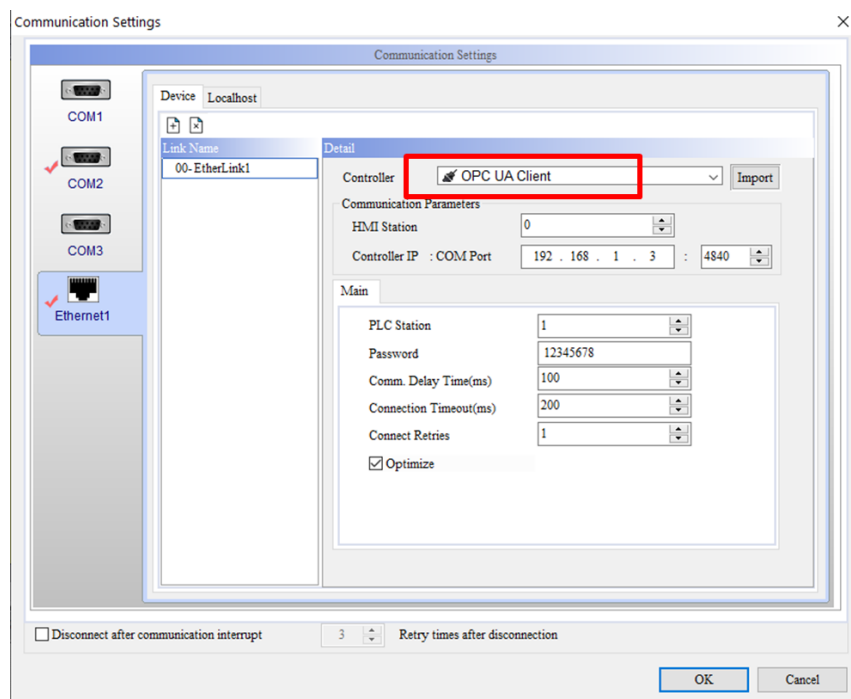


Загрузите проект в контроллер:

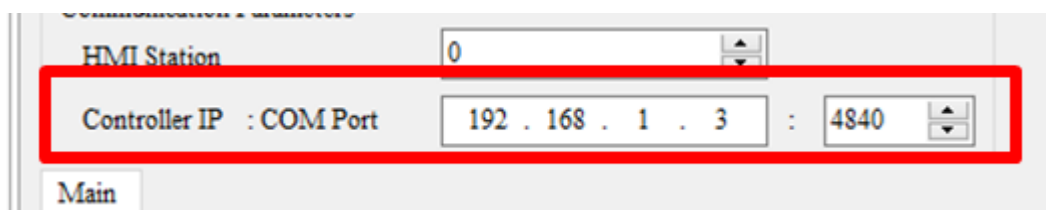


После загрузки проекта в контроллер, отмеченные переменные станут доступны для чтения-записи посредством протокола OPC UA.

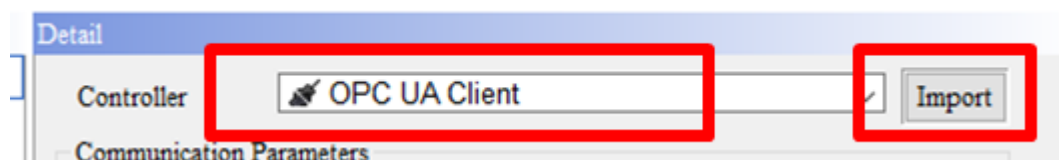
Для примера в качестве OPC UA клиента (Мастера) используем панель оператора Delta Electronics DOP-107WV. В среде программирования **DIAScreen** создайте проект для данной панели и в настройках порта Ethernet выберите драйвер **OPC UA Client**:



В качестве сервера укажите IP адрес контроллера MX308:

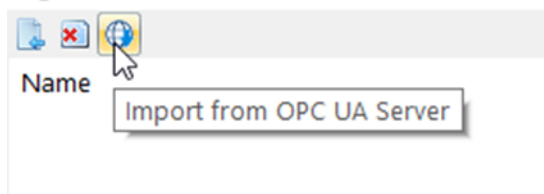


Далее нажмите кнопку **Import** справа от имени драйвера:



В открывшемся окне нажмите поиск сервера:

Tag List



При наличии нормального соединения ПК – контроллер, среда разработки DIAScreen найдёт в сети контроллер и выведет древо сервера (контроллер MX308):

Tag List

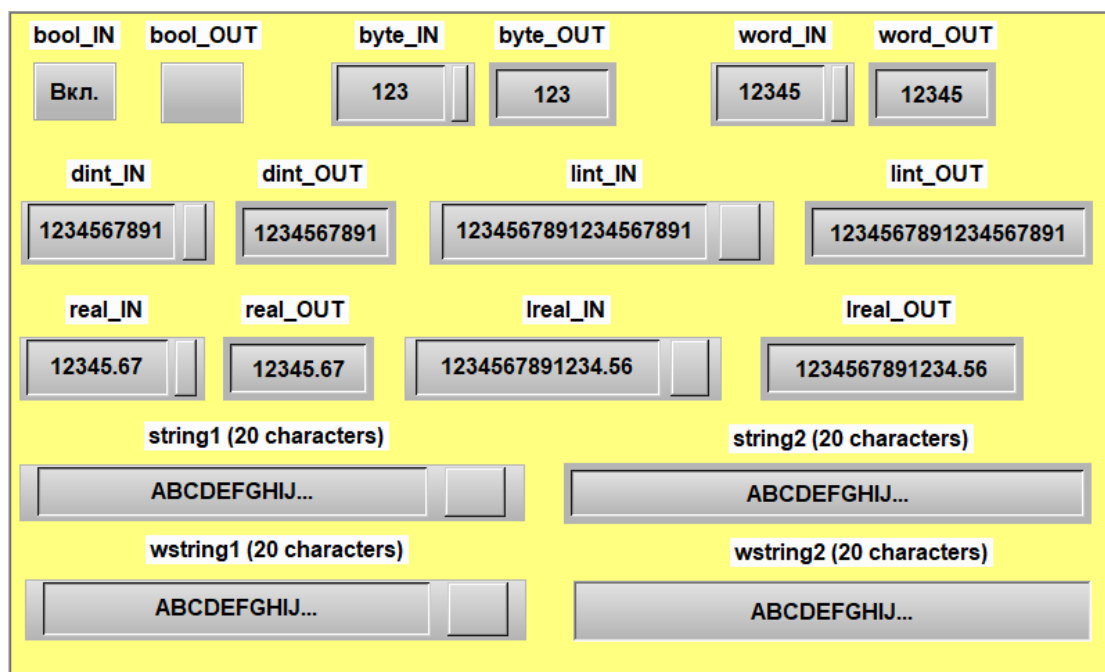
Name	Type	Node ID
Root		
Objects		
DeviceSet		ns=2;i=5001
Server		i=2253

Разверните полностью пункт **Server** до появления списка тегов:

Tag List

Name	Type	Node ID
Programs		ns=4;s= appo LS.Application.Program ^
POU		ns=4;s= var LS.Application.POU
bool_coil	Boolean	ns=4;s= var LS.Application.POU.bool_
bool_contact	Boolean	ns=4;s= var LS.Application.POU.bool_
byte_IN	Byte	ns=4;s= var LS.Application.POU.byte_I
byte_OUT	Byte	ns=4;s= var LS.Application.POU.byte_C
dint_IN	Int32	ns=4;s= var LS.Application.POU.dint_I
dint_OUT	Int32	ns=4;s= var LS.Application.POU.dint_C
lint_IN	Int64	ns=4;s= var LS.Application.POU.lint_I
lint_OUT	Int64	ns=4;s= var LS.Application.POU.lint_O
lreal_IN	Double	ns=4;s= var LS.Application.POU.lreal_I
lreal_OUT	Double	ns=4;s= var LS.Application.POU.lreal_C
real_IN	Float	ns=4;s= var LS.Application.POU.real_I
real_OUT	Float	ns=4;s= var LS.Application.POU.real_C
string1	String	ns=4;s= var LS.Application.POU.string
string2	String	ns=4;s= var LS.Application.POU.string
word_IN	UInt16	ns=4;s= var LS.Application.POU.word_
word_OUT	UInt16	ns=4;s= var LS.Application.POU.word_
wstring1	String	ns=4;s= var LS.Application.POU.wstrin
wstring2	String	ns=4;s= var LS.Application.POU.wstrin
Tasks		ns=4;s= lannoll S Application Tasks

Для демонстрации передачи данных достаточно создать несколько экранных объектов в соответствии с типами данных.



Для тегов типа BOOL используются объекты типа Button Maintained и Multistate Indicator. Числовые типы данных отображаются и вводятся через объекты типа Numeric Input/Display, которые настраиваются на соответствующий тип данных. Например, для типа WORD:

HardwareRevision	String	ns=4;s= vprop LS.Applic
Programs		ns=4;s= appo LS.Applica
POU		ns=4;s= var LS.Applicati
byte_IN	Byte	ns=4;s= var LS.Applicati
byte_OUT	Byte	ns=4;s= var LS.Applicati
dint_IN	Int32	ns=4;s= var LS.Applicati
dint_OUT	Int32	ns=4;s= var LS.Applicati
lint_IN	Int64	ns=4;s= var LS.Applicati
lint_OUT	Int64	ns=4;s= var LS.Applicati
lreal_IN	Double	ns=4;s= var LS.Applicati
lreal_OUT	Double	ns=4;s= var LS.Applicati
real_IN	Float	ns=4;s= var LS.Applicati
real_OUT	Float	ns=4;s= var LS.Applicati
string1	String	ns=4;s= var LS.Applicati
string2	String	ns=4;s= var LS.Applicati
word_IN	UInt16	ns=4;s= var LS.Applicati
word_OUT	UInt16	ns=4;s= var LS.Applicati
wstring1	String	ns=4;s= var LS.Applicati
wstring2	String	ns=4;s= var LS.Applicati

необходимо сделать следующие настройки:

Detail

Data Type Word

Data Format Unsigned Decimal

☒ System Keypad

☐ Custom Keypad

Minimum 0

Maximum 65535

Integer Digits 5

Fractional 0

Min 0

Max 65535

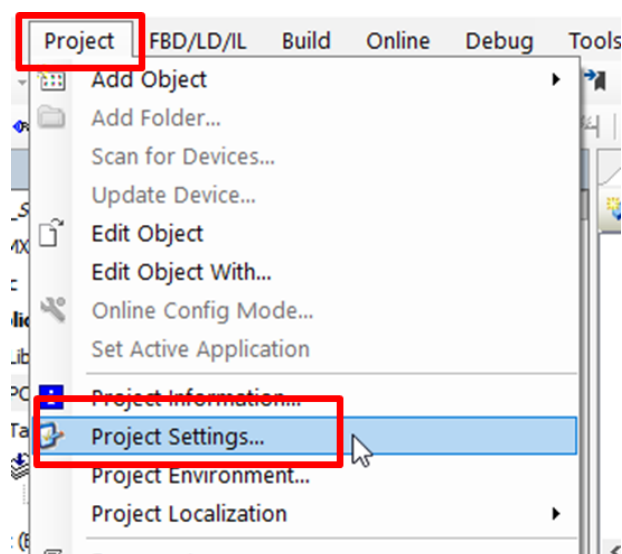
Для ввода и отображения символов типа ASCII (1 байт) используются объекты Character Input/Display. В контроллере данный тип определяется как `string(**)`. В скобках указывается количество символов (байтов).

В настройках объектов панели количество символов указывается аналогичное (байты).

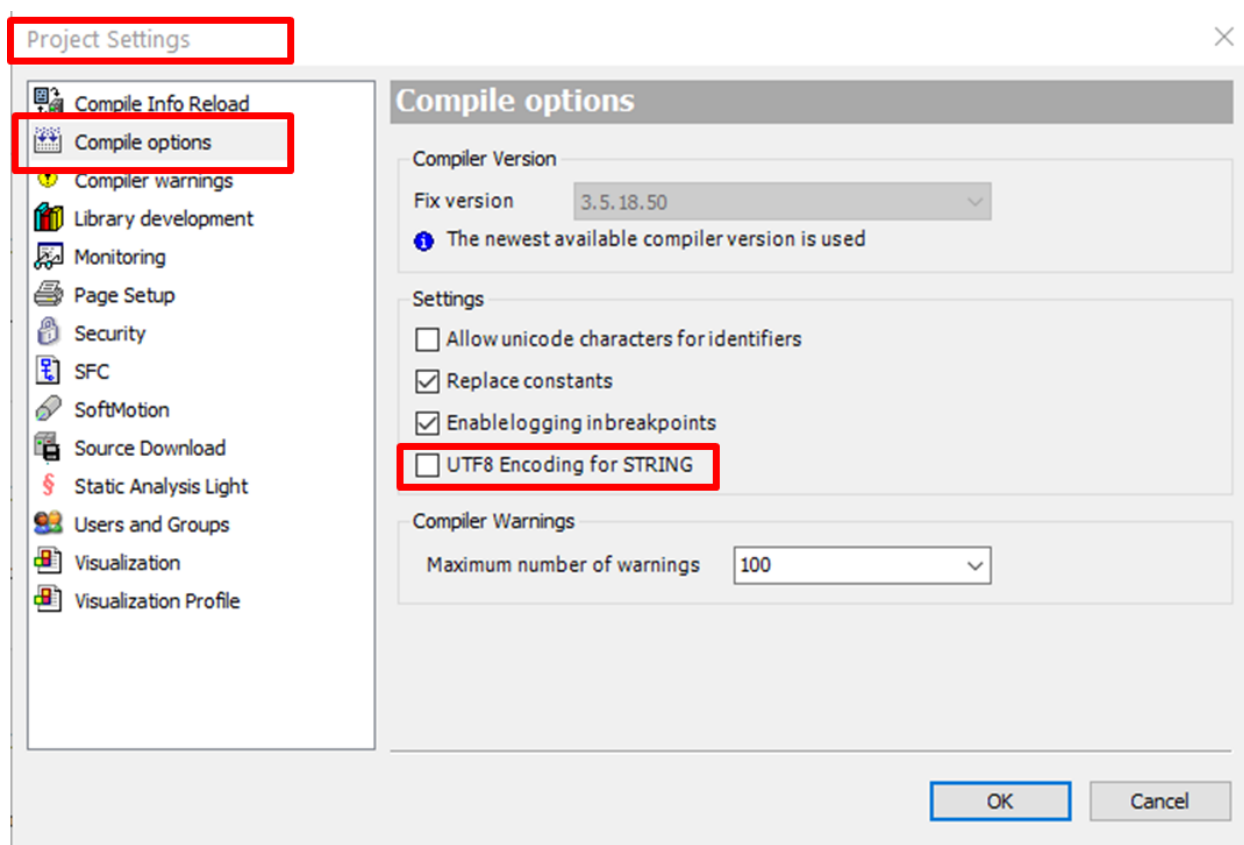
Для ввода и отображения символов типа Unicode (2 байта) используется объект Multilanguage Input. В контроллере данный тип определяется как `wstring(**)`. В скобках указывается количество символов. В типе данных `wstring` используется кодировка типа UCS2, в которой каждый символ занимает 2 байта.

Для однозначности трактования типов `string` и `wstring` в настройках среды программирования контроллера необходимо убедиться в наличии следующих настроек:

Project – Project Settings:



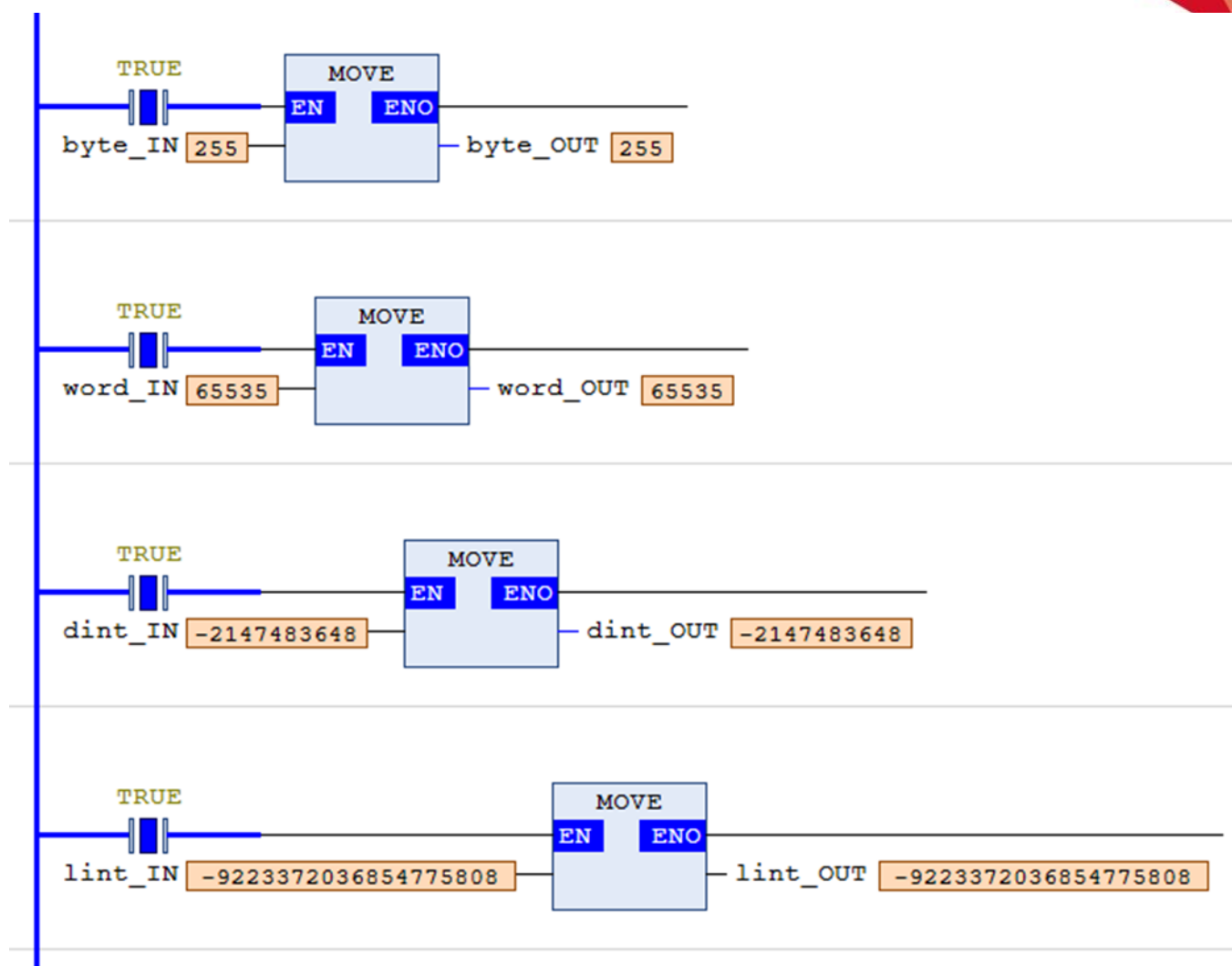
Флажок **UTF8 Encoding for String** должен быть снят!



Для наглядности в проекте контроллера можно поставить созданные теги в тело программы или вывести в таблицу мониторинга Watch Table.

Загрузите проекты в контроллер и в панель. Введите значения в переменные. Они будут отображаться одинаково и в панели и в контроллере. В нашем примере будет выглядеть так:




Watch 1

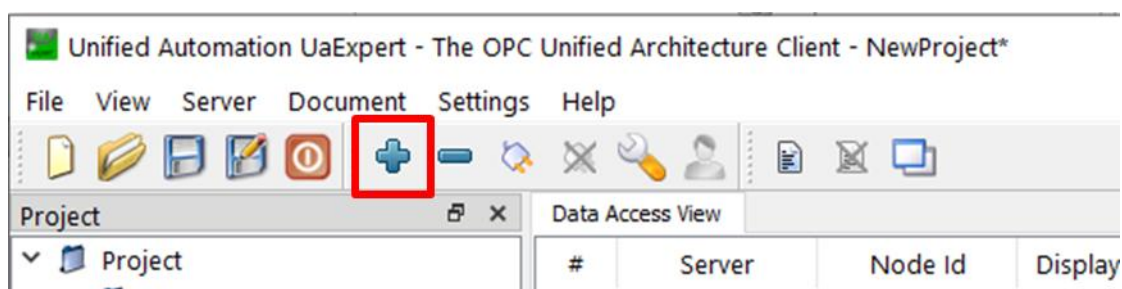
Expression	Type	Value
POU.wstring2	WSTRING(20)	"А5ВГγke123@#\$\$%&<=np0"
POU.string2	STRING(20)	'ABCDdsw!@#\$\$%^&*()123'
POU.real_OUT	REAL	-99999.99
POU.lreal_OUT	LREAL	-99999999999999.99

bool_IN Выкл.	bool_OUT 	byte_IN 255	byte_OUT 255	word_IN 65535	word_OUT 65535
dint_IN -2147483648	dint_OUT -2147483648	lint_IN -9223372036854775808		lint_OUT -9223372036854775808	
real_IN -99999.99	real_OUT -99999.99	lreal_IN -99999999999999.99		lreal_OUT -99999999999999.99	
string1 (20 characters) ABCDdsw!@#\$\$%^&*()123			string2 (20 characters) ABCDdsw!@#\$\$%^&*()123		
wstring1 (20 characters) АБВГук123@#\$\$%&<=про			wstring2 (20 characters) АБВГук123@#\$\$%&<=про		

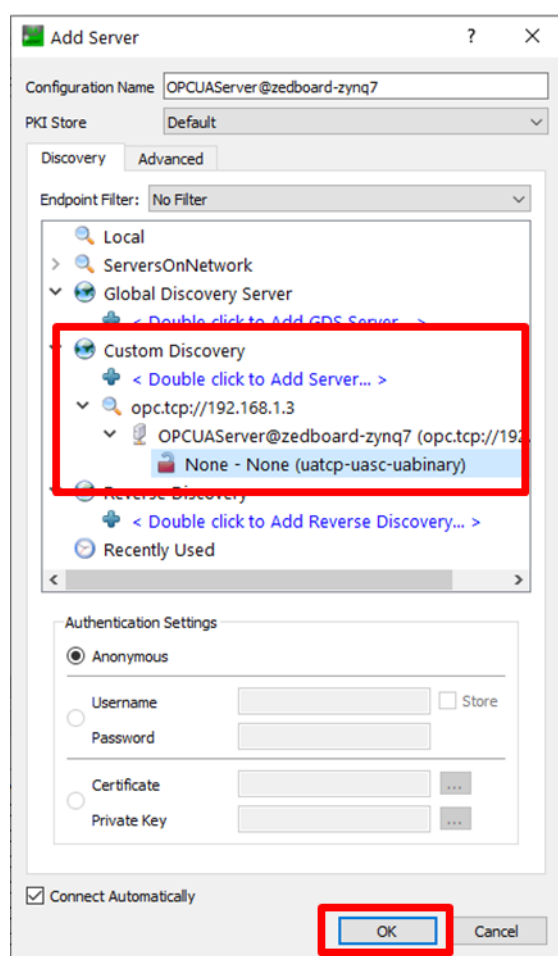
Для демонстрации сделано так, что через переменные типа *****_IN** данные вводятся, а через переменные типа *****_OUT** эти же данные выводятся.

Также, посмотреть работу OPC UA Server в контроллере можно через стандартную утилиту **UaExpert**. Для этого необходимо иметь связь ПК – контроллер, запустить утилиту, добавить сервер, установить связь с ним, выбрать переменные, с которыми необходимо работать.

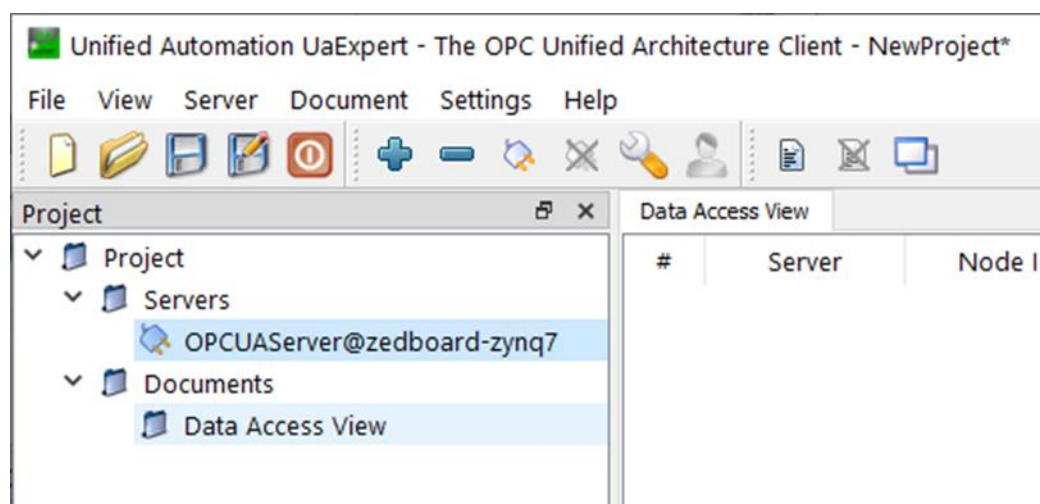
Добавление сервера:



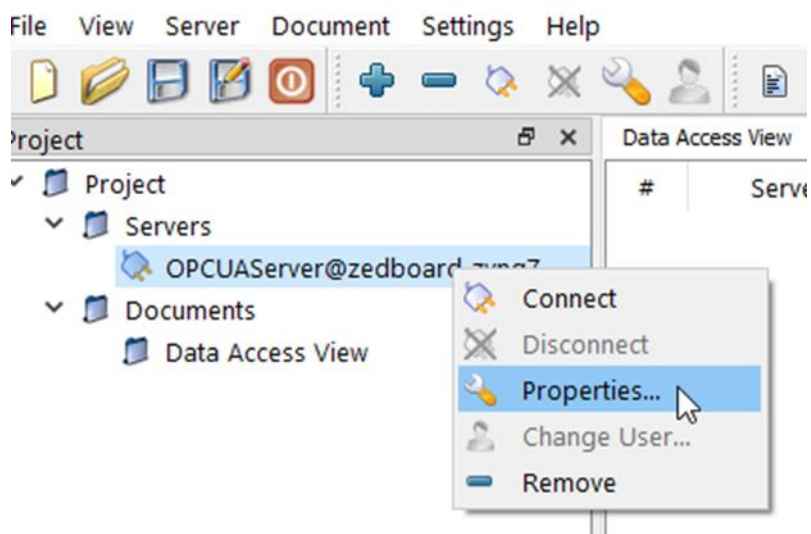
В открывшемся окне выберите сервер (здать IP адрес) и нажать **ОК**:



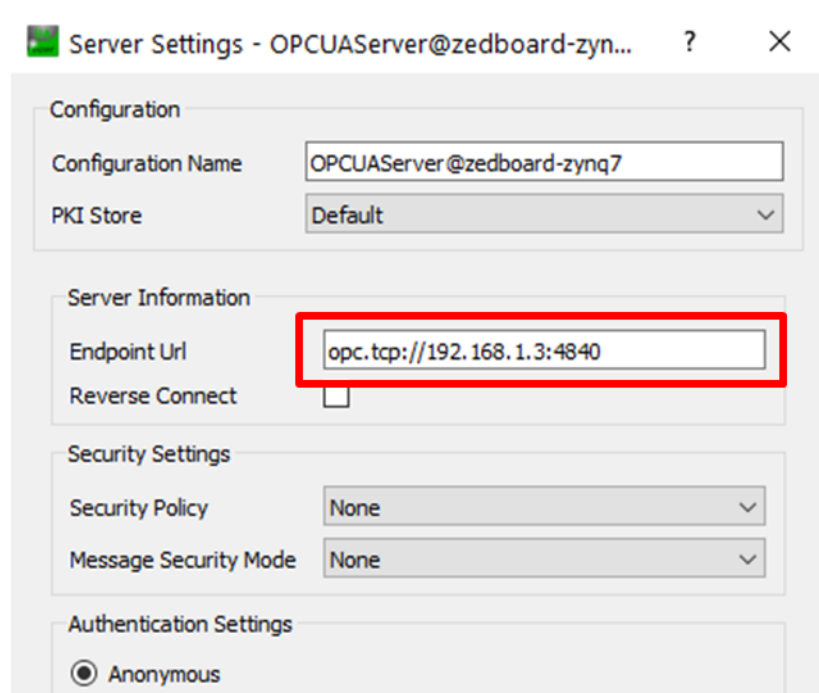
В древе проекта появится сервер:



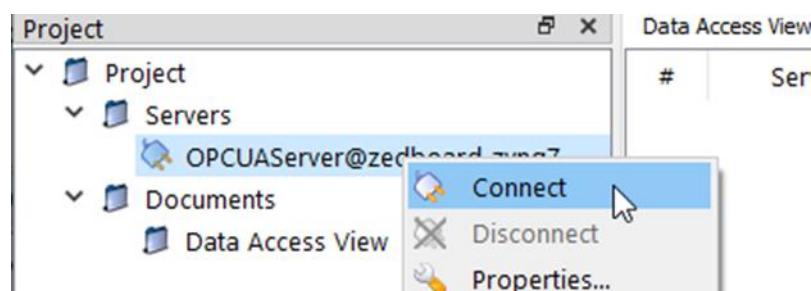
Щёлкните правой кнопкой мышки на названии сервера и выберите пункт **Properties**:



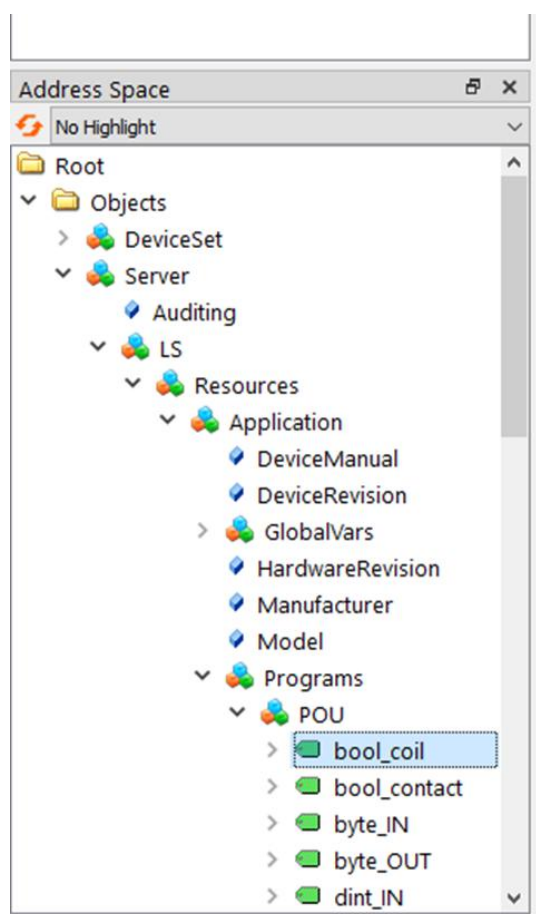
Введите IP адрес сервера (контроллера) и нажмите **OK**:



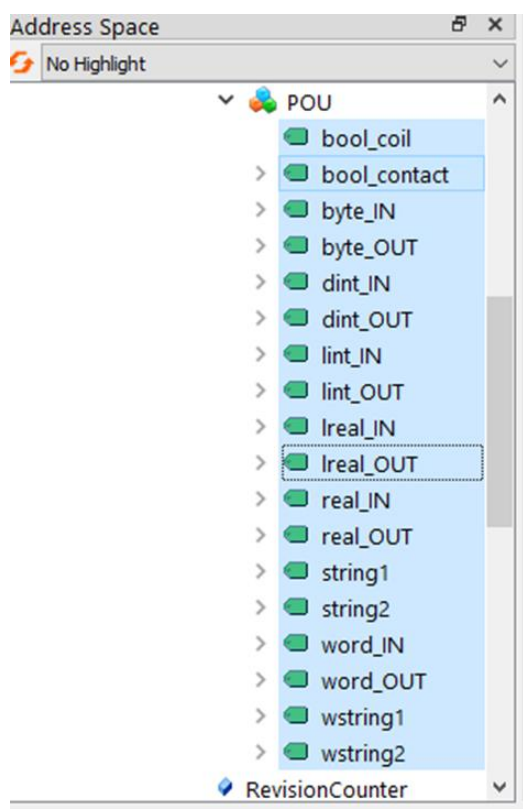
Ещё раз нажмите правой кнопкой на имени сервера и нажмите пункт **Connect**:



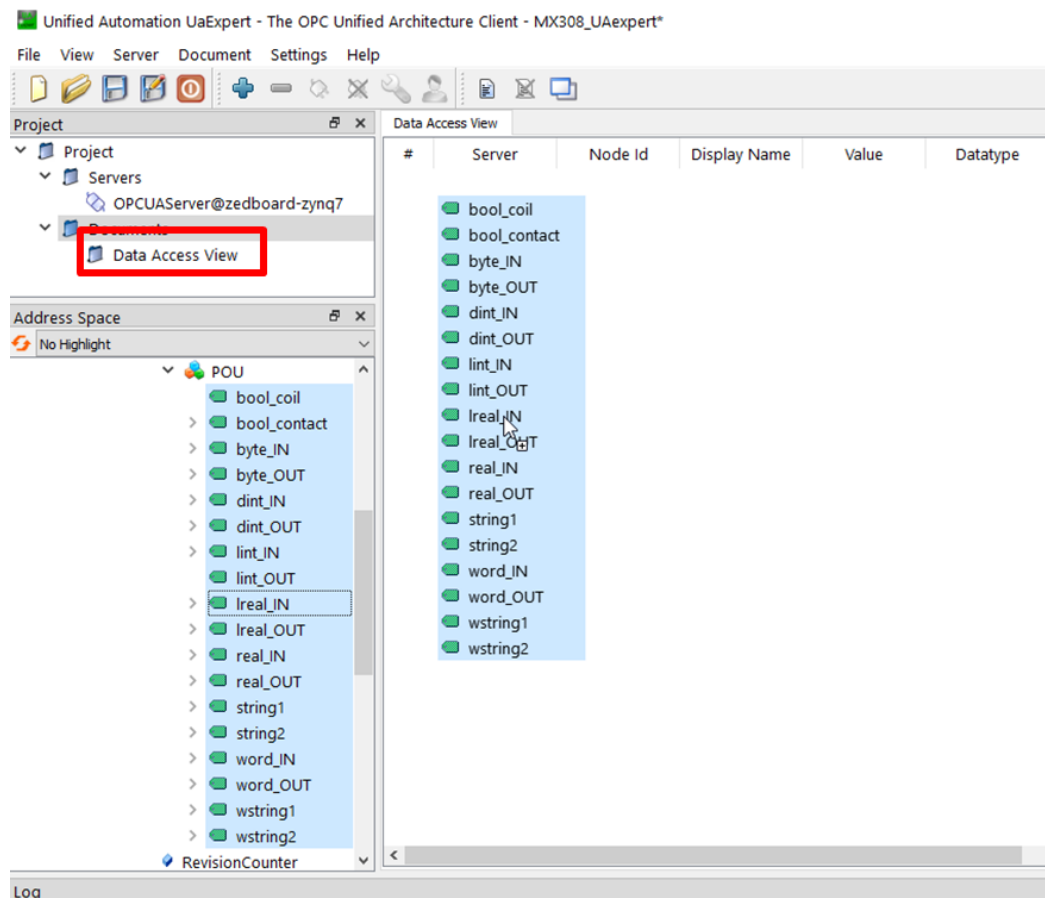
Откроется древо сервера, развернув которое Вы увидите теги:



Выделите нужные теги:



Далее откройте вкладку **Data Access View** и мышкой по технологии DRAG&DROP переместите теги в область мониторинга:



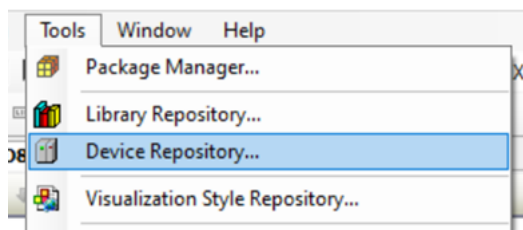
Далее появятся данные, которые в колонке Value можно менять (если переменные были помечены как Read-Write при создании).

Data Access View						
#	Display Name	Value	Datatype	Source Timestamp	Server Timestamp	Status
1	bool_coil	true	Boolean	18:53:58.489	18:53:58.489	Good
2	bool_contact	true	Boolean	18:53:58.489	18:53:58.489	Good
3	byte_IN	255	Byte	18:53:58.489	18:53:58.489	Good
4	byte_OUT	255	Byte	18:53:58.489	18:53:58.489	Good
5	dint_IN	-2147483648	Int32	18:53:58.489	18:53:58.489	Good
6	dint_OUT	-2147483648	Int32	18:53:58.489	18:53:58.489	Good
7	lint_IN	-9223372036854775808	Int64	18:53:58.489	18:53:58.489	Good
8	lint_OUT	-9223372036854775808	Int64	18:53:58.489	18:53:58.489	Good
9	string1	ABCDdsw!@#\$%^&*()123	String	18:53:58.489	18:53:58.489	Good
10	string2	ABCDdsw!@#\$%^&*()123	String	18:53:58.489	18:53:58.489	Good
11	word_IN	65535	UInt16	18:53:58.489	18:53:58.489	Good
12	word_OUT	65535	UInt16	18:53:58.489	18:53:58.489	Good
13	wstring1	АБВГыке123@#\$%^&*()123	String	18:53:58.489	18:53:58.489	Good
14	wstring2	АБВГыке123@#\$%^&*()123	String	18:53:58.489	18:53:58.489	Good

Работа с преобразователем частоты серии Optimus Drive AD800 по сети EtherCAT

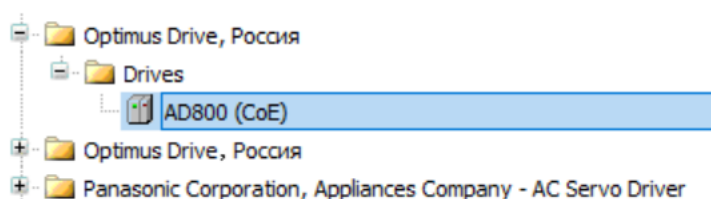
Преобразователи частоты (ПЧ) серии **AD800** имеют возможность расширяться функциональными платами. Для интеграции ПЧ в сеть EtherCAT используется плата **ET01**. Установка и работа с данной платой приведены в Руководстве по эксплуатации «Плата расширения EtherCAT ET01».

Для включения ПЧ в состав проекта контроллера серии МН1048 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,

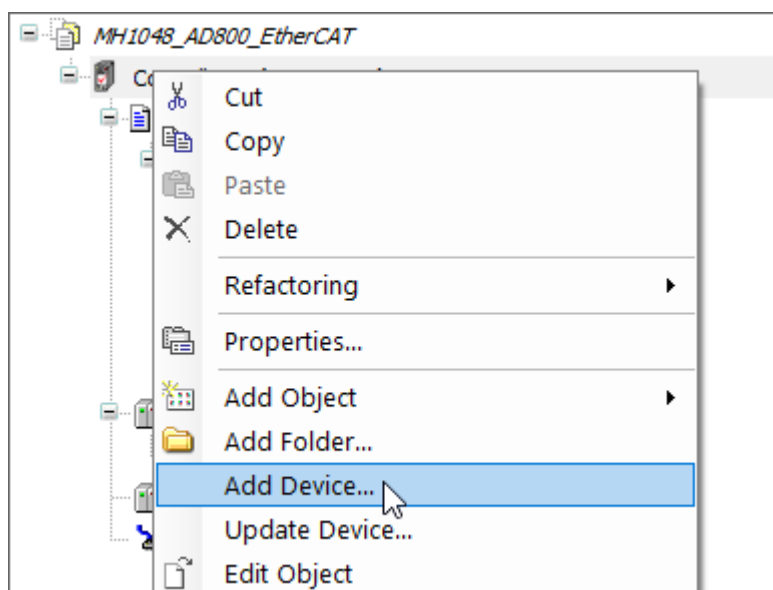
P0-11 = 20,

P0-17 = 0,

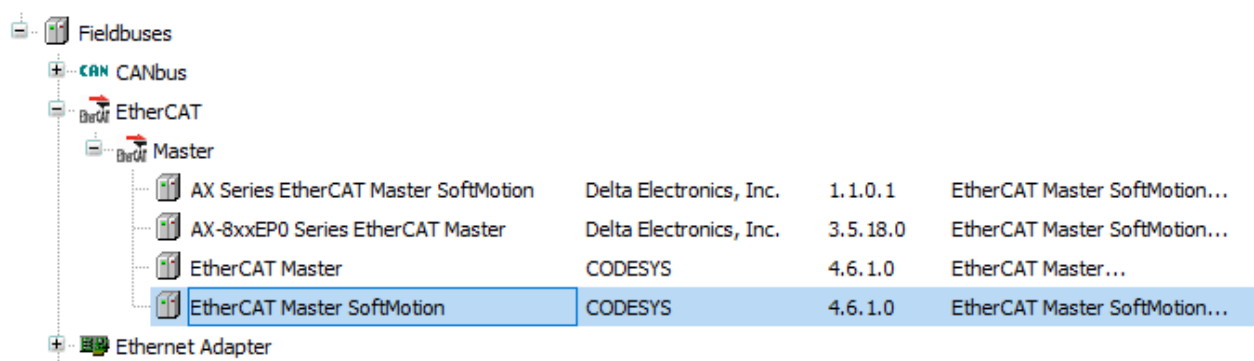
P0-18 = 2,

P0-29 = 1

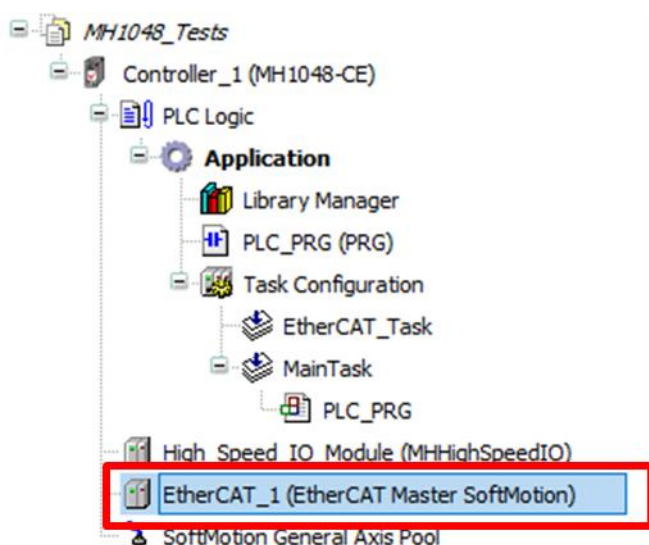
Далее необходимо добавить в проект EtherCAT адаптер. Для этого щёлкните правой кнопкой на названии контроллера и в меню выберите пункт **Add Device**:



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

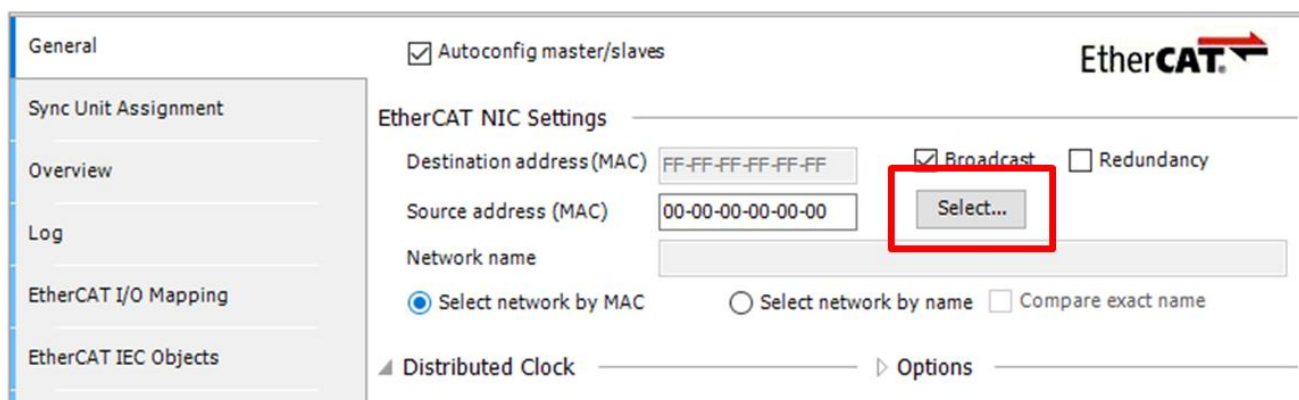


В древе проекта появится пункт:



Далее необходимо назначить в проекте Мастера сети EtherCAT. В нашем примере это будет контроллер МН1048, 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**:



General ☒ Autoconfig master/slaves **EtherCAT**

EtherCAT NIC Settings

Destination address (MAC) FF-FF-FF-FF-FF-FF ☒ Broadcast ☐ Redundancy

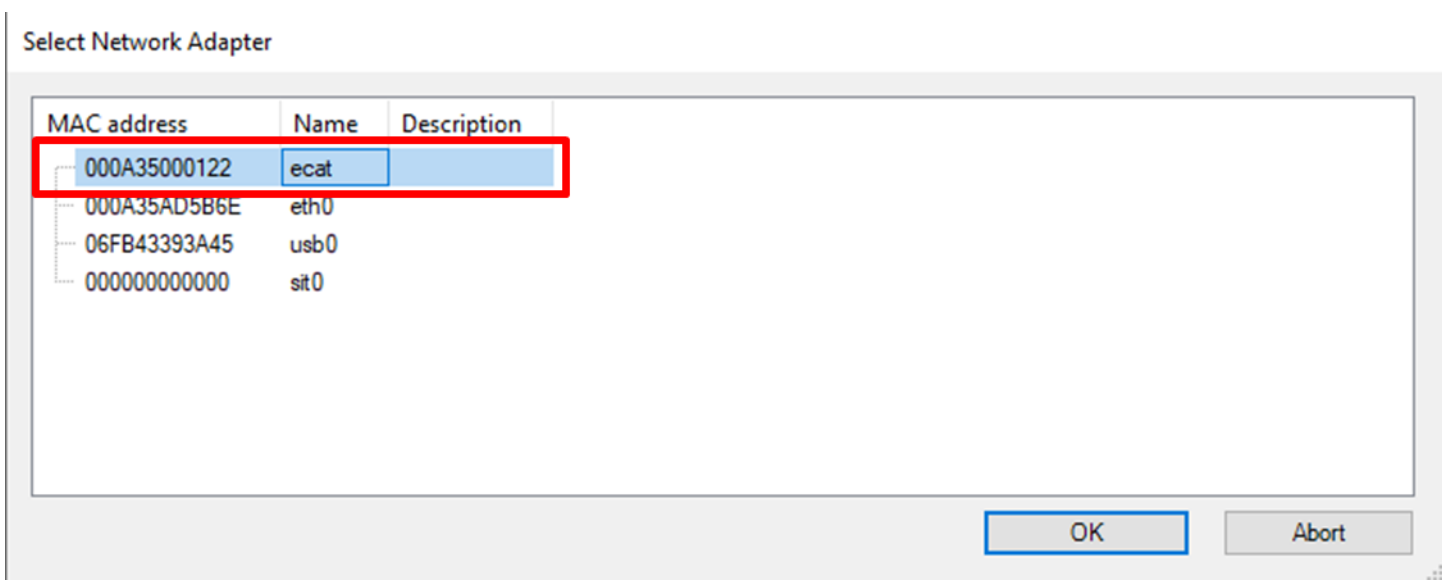
Source address (MAC) 00-00-00-00-00-00 **Select...**

Network name

☒ Select network by MAC ☐ Select network by name ☐ Compare exact name

☒ Distributed Clock ☐ Options

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

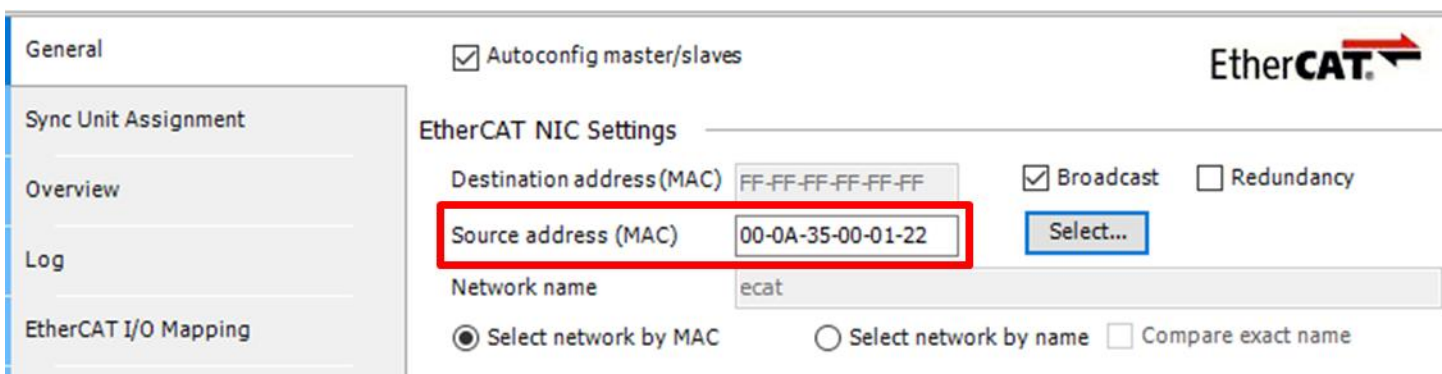


Select Network Adapter

MAC address	Name	Description
000A35000122	ecat	
000A35AD5B6E	eth0	
06FB43393A45	usb0	
000000000000	sit0	

OK Abort

MAC адрес должен установиться в данном поле:



General ☒ Autoconfig master/slaves **EtherCAT**

EtherCAT NIC Settings

Destination address (MAC) FF-FF-FF-FF-FF-FF ☒ Broadcast ☐ Redundancy

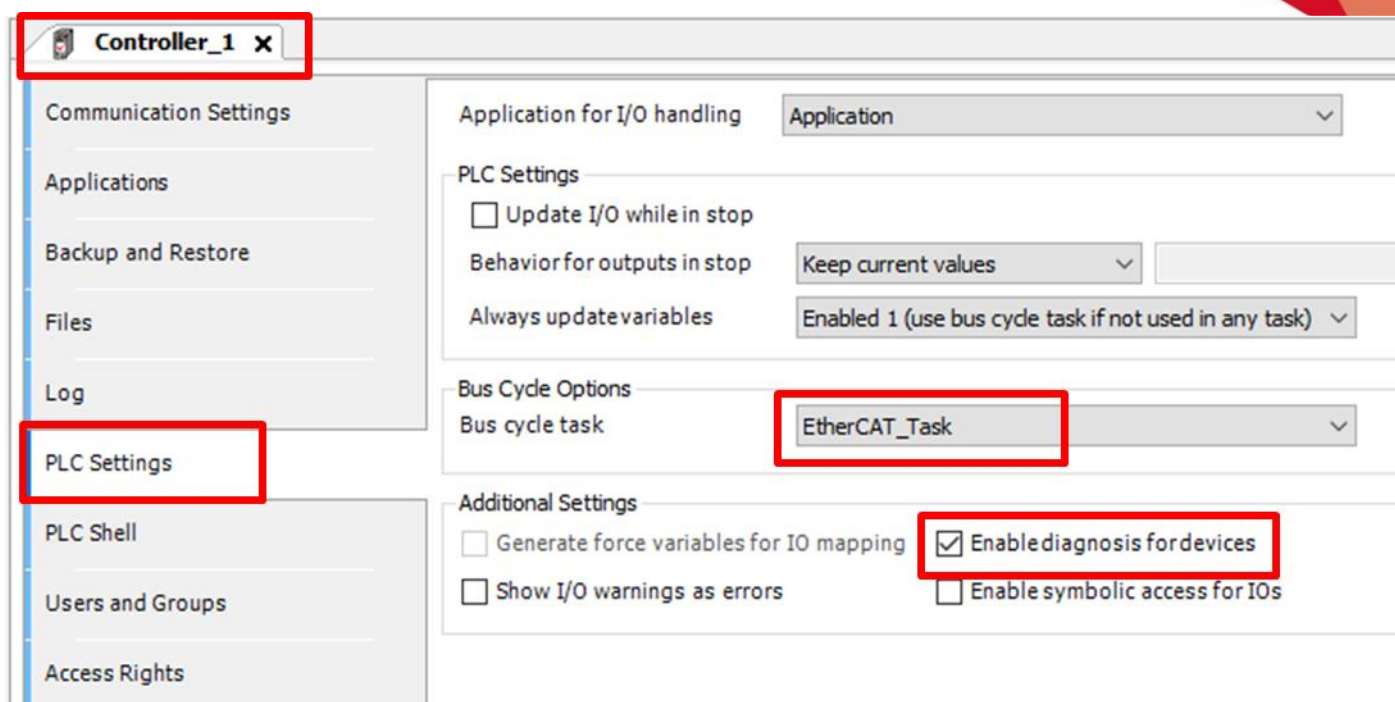
Source address (MAC) 00-0A-35-00-01-22 **Select...**

Network name ecat

☒ Select network by MAC ☐ Select network by name ☐ Compare exact name

Далее необходимо активировать задачу EtherCAT_Task. Для этого сделайте следующие настройки:

Во вкладке Контроллера:



Controller_1 X

- Communication Settings
- Applications
- Backup and Restore
- Files
- Log
- PLC Settings**
- PLC Shell
- Users and Groups
- Access Rights

Application for I/O handling: Application

PLC Settings

☐ Update I/O while in stop

Behavior for outputs in stop: Keep current values

Always update variables: Enabled 1 (use bus cycle task if not used in any task)

Bus Cycle Options

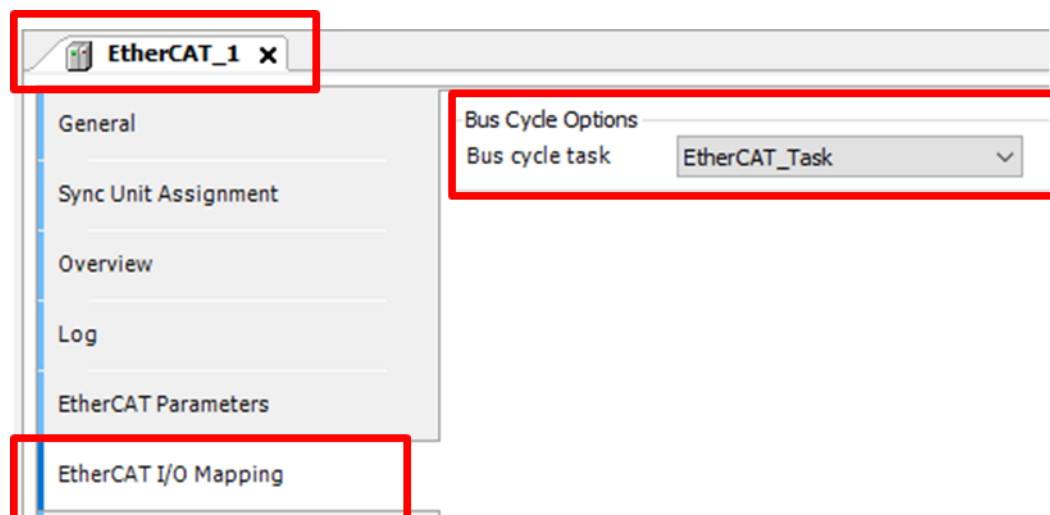
Bus cycle task: **EtherCAT_Task**

Additional Settings

☐ Generate force variables for IO mapping ☒ **Enabled diagnosis for devices**

☐ Show I/O warnings as errors ☐ Enable symbolic access for IOs

Во вкладке EtherCAT_Master:



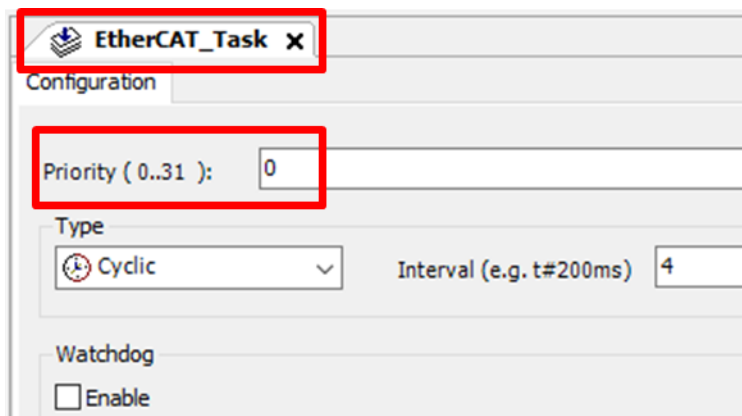
EtherCAT_1 X

- General
- Sync Unit Assignment
- Overview
- Log
- EtherCAT Parameters
- EtherCAT I/O Mapping**

Bus Cycle Options

Bus cycle task: EtherCAT_Task

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



EtherCAT_Task X

Configuration

Priority (0..31): 0

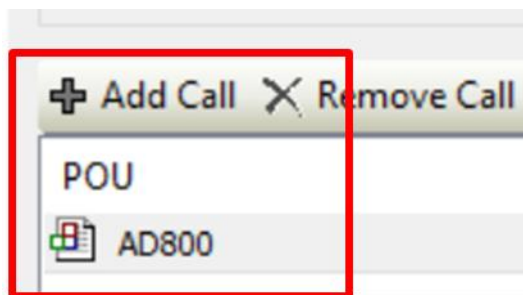
Type: Cyclic

Interval (e.g. t#200ms): 4

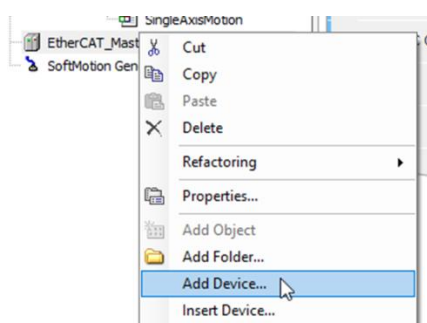
Watchdog

☐ Enable

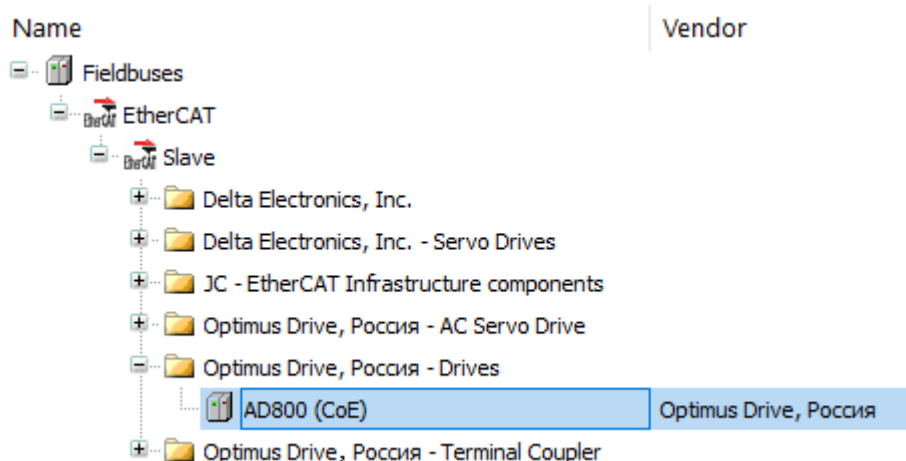
Для приводов AD800 рекомендуется такт шины не менее 4 мс. При большом количестве приводов на шине время опроса необходимо увеличить.



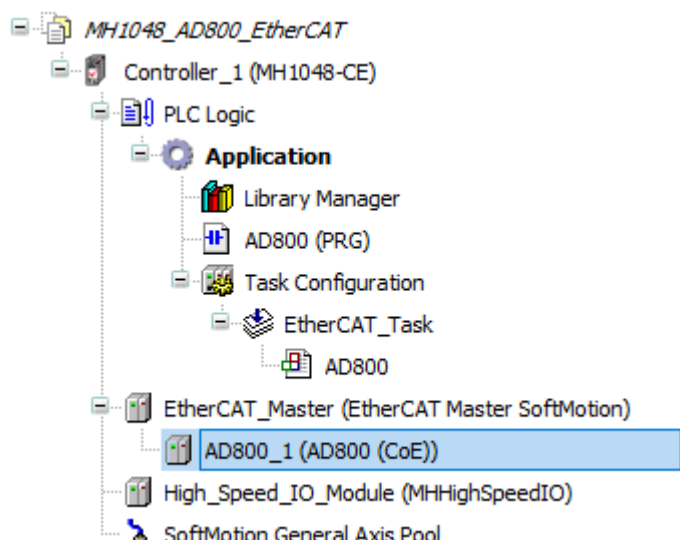
Далее необходимо добавить в проект привод. Для этого щёлкните правой кнопкой мышки на пункте **EtherCAT Master** и в открывшемся окне выберите пункт **Add Device**:



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		
		Name	Type	Index			
Process Data		☒ 16#1610 Outputs			☒ 16#1A10 Inputs		
		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
Log							

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

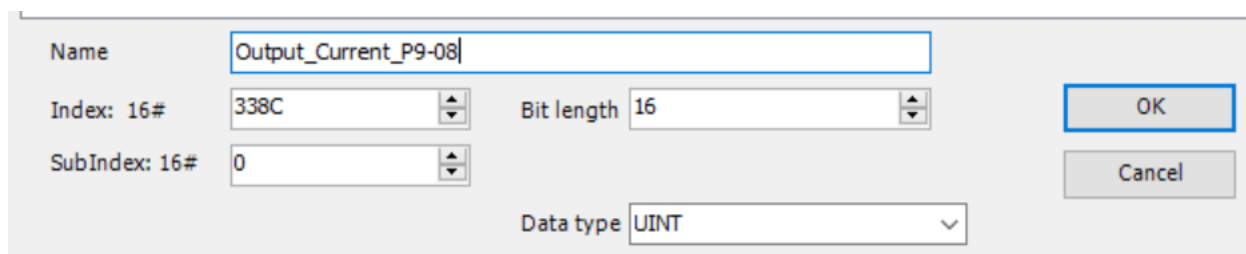
В разделе **Outputs**, т.е. данные в сторону привода, будет два регистра – **ControlWord** (команды приводу) и **Set_Value** (задание скорости в Герцах).

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:

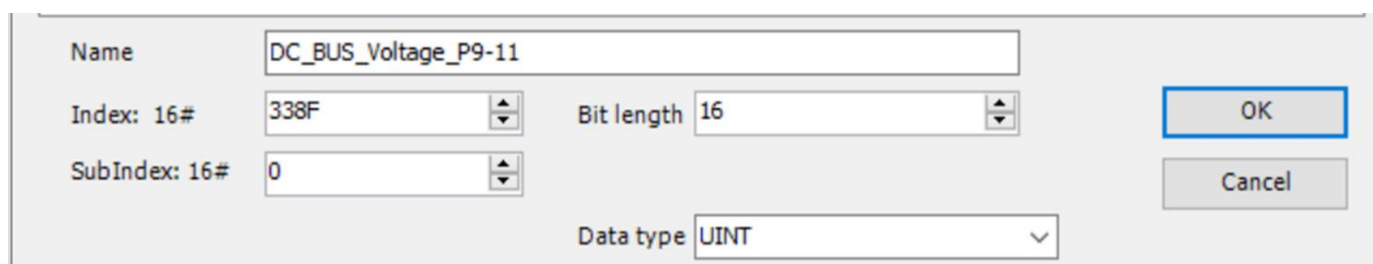
General	Address	Additional
Expert Process Data Process Data	AutoInc address 0 EtherCAT address 1001	☒ Expert settings ☐ Optional

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 UNIT:



В итоге будет список параметров:

PDO Content (16#1A10)			
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:

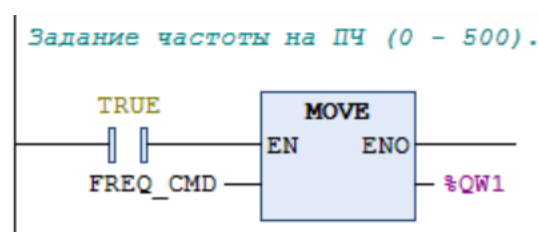
General	Find Filter Show all				
	Variable	Mapping	Channel	Address	Type
Expert Process Data	16#1610 Outputs				
Process Data	  		ControlWord	%QW0	UINT
Log	  		Set_Value	%QW1	UINT
EtherCAT Parameters	16#1A10 Inputs				
EtherCAT I/O Mapping	  		Status_Word_P9-01	%IW0	UINT
	  		Output_Frequency_P9-07	%IW1	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:



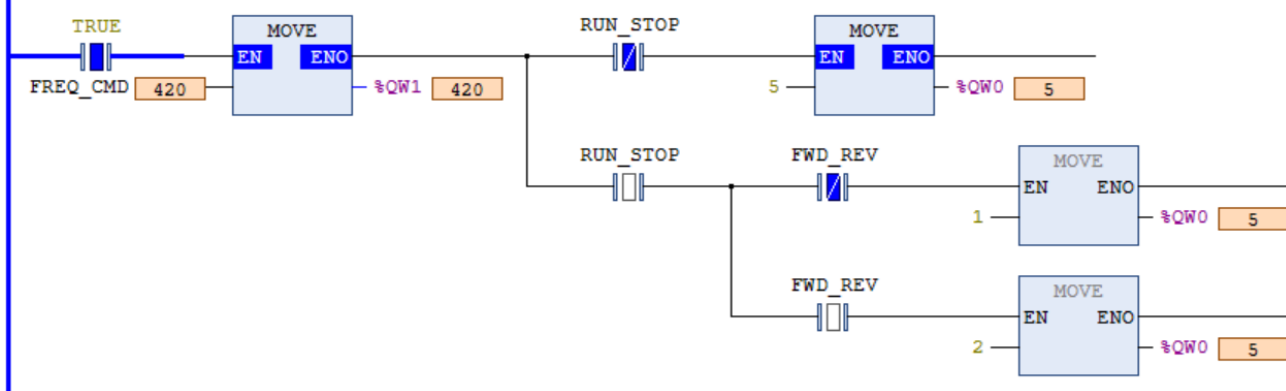
Для управления приводом в контрольное слово нужно записать следующие значения:

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

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

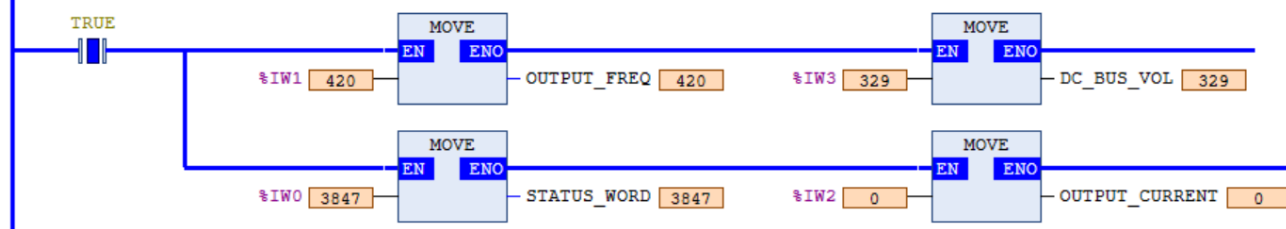
Для удобства работы команды можно описать в программе через булевы переменные, которые будут отправлять нужное число в командный регистр, например такой код:

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:

Мониторинг состояния ПЧ.



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**.

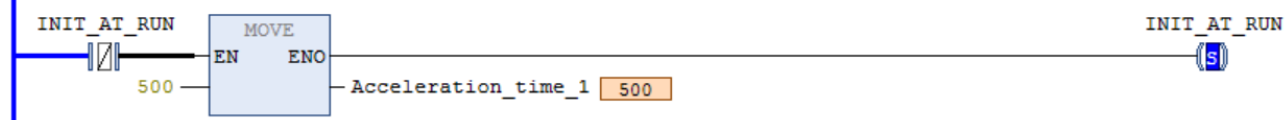
Для примера запишем в привод и для контроля прочитаем 1-е время разгона, это параметр P0-51. Адрес для EtherCAT вычисляется также как для пакетов PDO (см. выше по тексту).

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

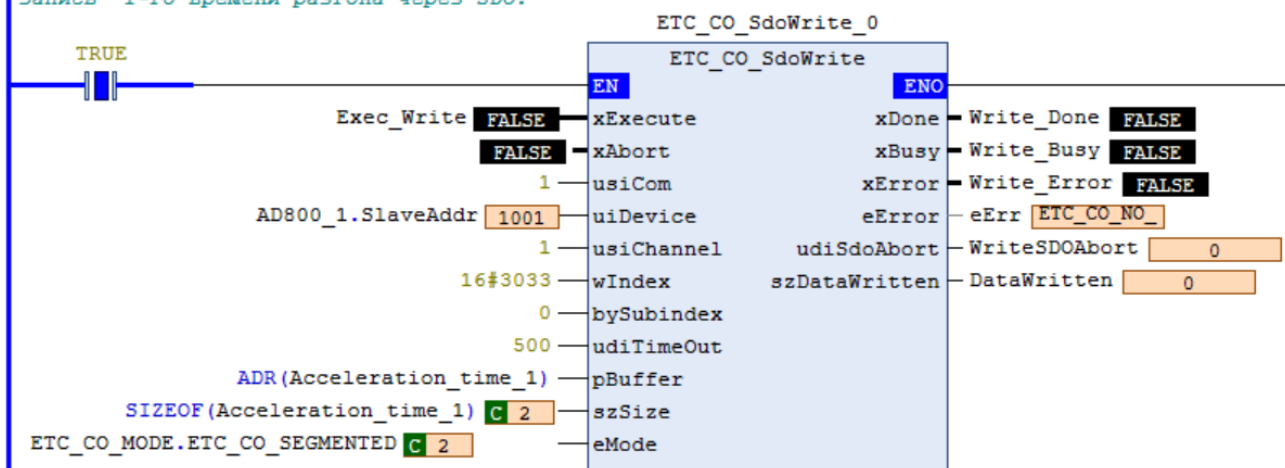
Блок записи значения времени разгона в привод:

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*.



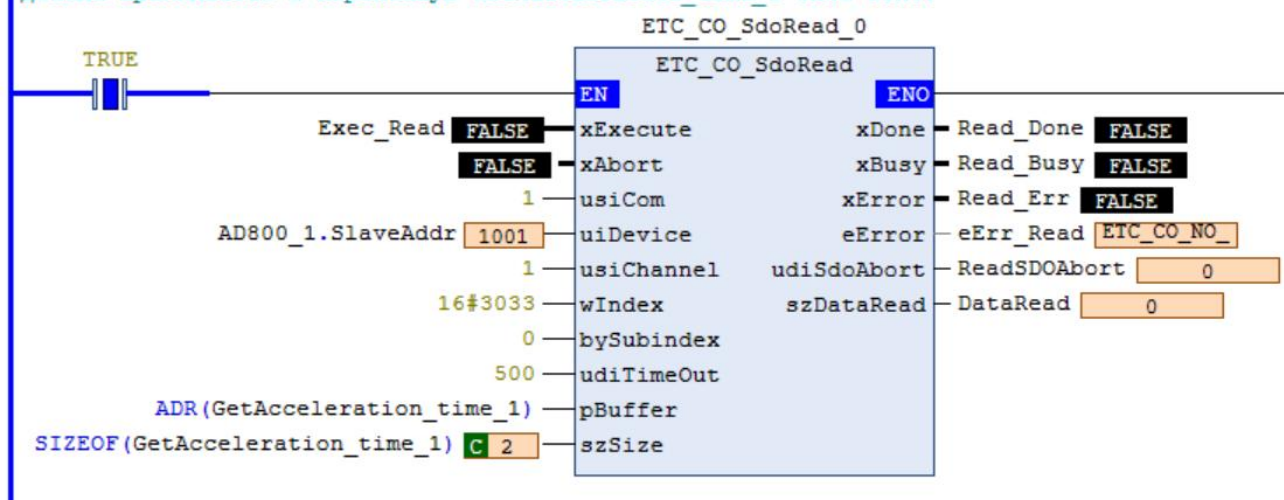
Запись 1-го времени разгона через SDO.



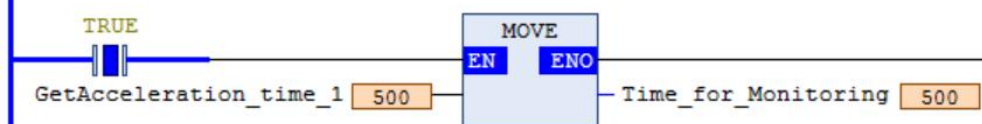
Блок чтения значения времени разгона из привода:

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.



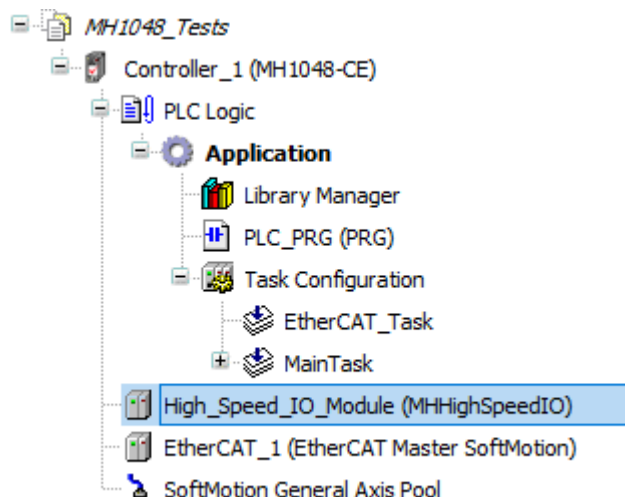
Для мониторинга.



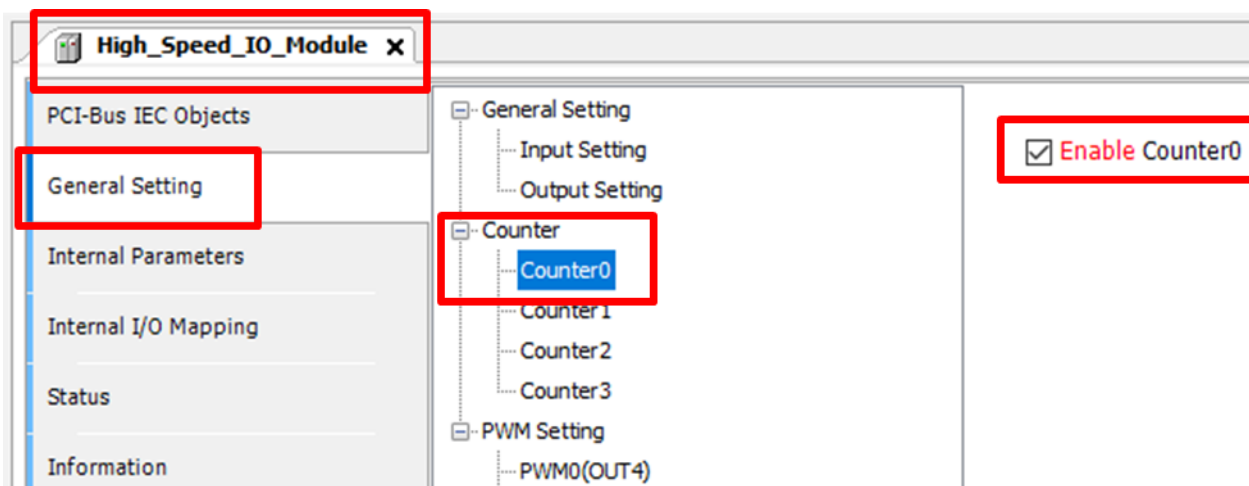
Функция Touch Probe

Контроллеры серии MH1048 поддерживают функцию захвата значения высокоскоростного счётчика по переднему фронту сигнала на входе контроллера. В мировой практике данная функция называется **Touch Probe**. Поддерживается для всех 4-х счётчиков **Counter0** – **Counter3**.

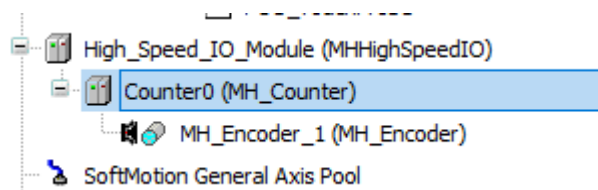
Для реализации функции захвата в древо проекта контроллера должен быть добавлен адаптер входов-выходов на ЦПУ – **MHHighSpeedIO** (см. соответствующую Главу данного Руководства). Это необходимо для организации высокоскоростного счётчика, текущее значение которого и будет захватываться.



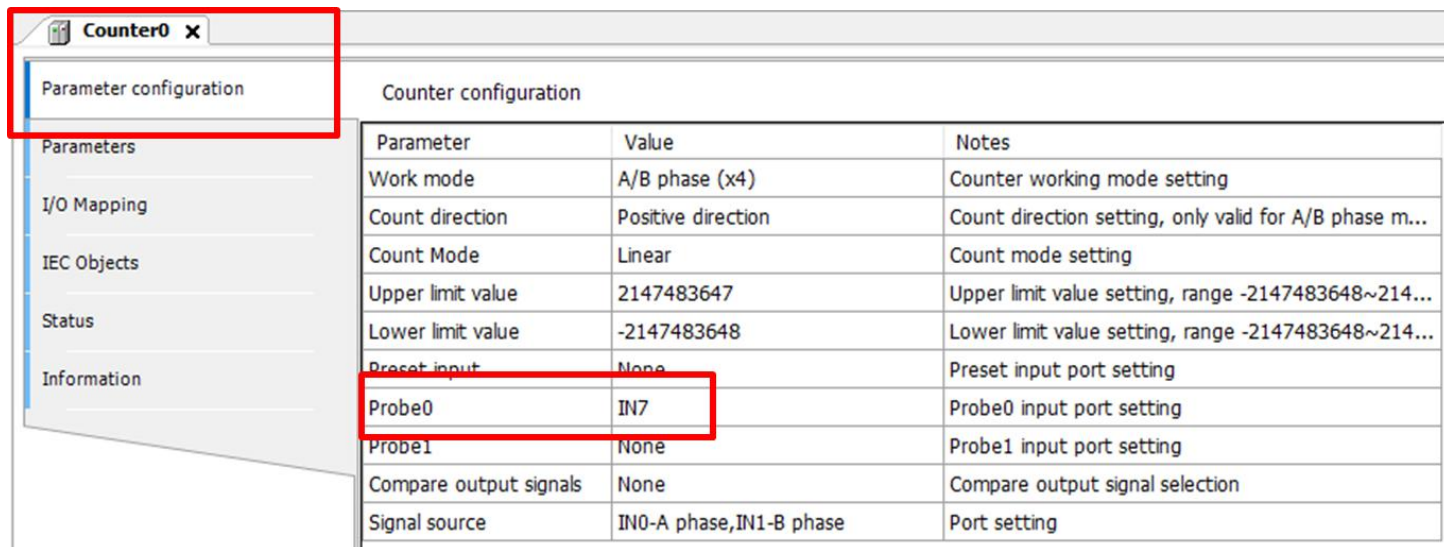
Создайте счётчик **Counter0** (см. Работа с высокоскоростными счётчиками, стр. 84 настоящего Руководства). Откройте вкладку **MHHighSpeedIO – General Setting – Counter0**.



В древе проекта появится счётчик и ось:

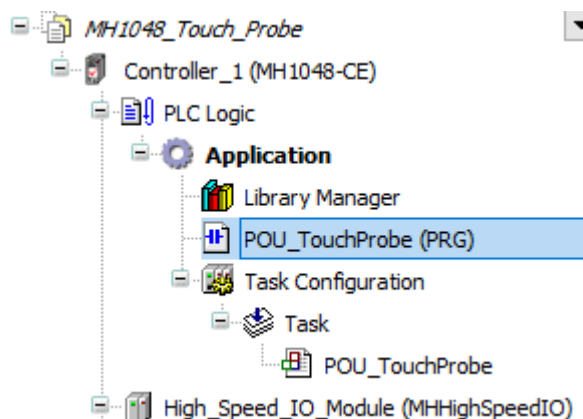


Двойным щелчком левой кнопки мышки на пункте **Counter0**, откроется вкладка настройки счётчика, назначьте вход для функции TouchProbe (в примере ниже IN7):

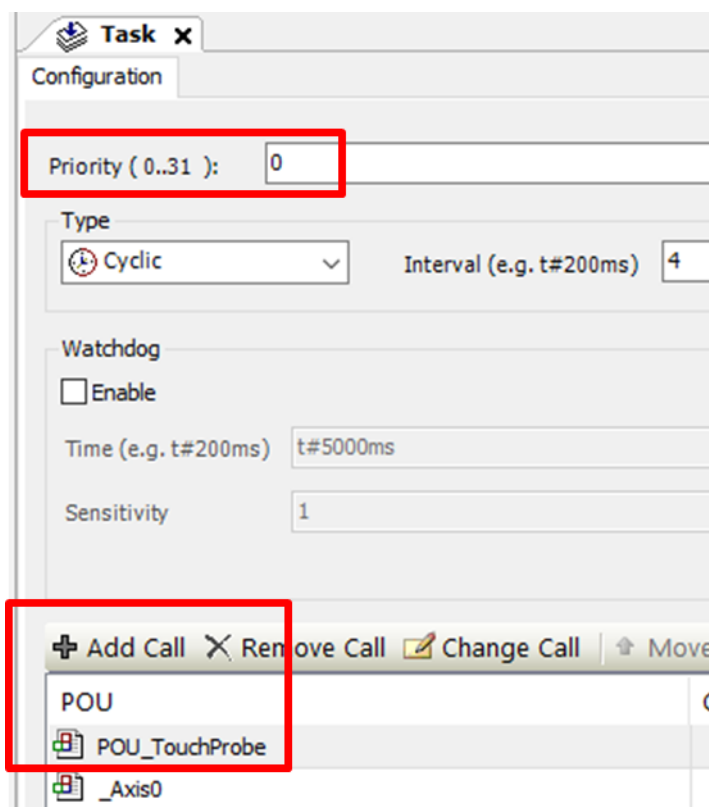


Parameter	Value	Notes
Work mode	A/B phase (x4)	Counter working mode setting
Count direction	Positive direction	Count direction setting, only valid for A/B phase m...
Count Mode	Linear	Count mode setting
Upper limit value	2147483647	Upper limit value setting, range -2147483648~214...
Lower limit value	-2147483648	Lower limit value setting, range -2147483648~214...
Preset input	None	Preset input port setting
Probe0	IN7	Probe0 input port setting
Probe1	None	Probe1 input port setting
Compare output signals	None	Compare output signal selection
Signal source	IN0-A phase, IN1-B phase	Port setting

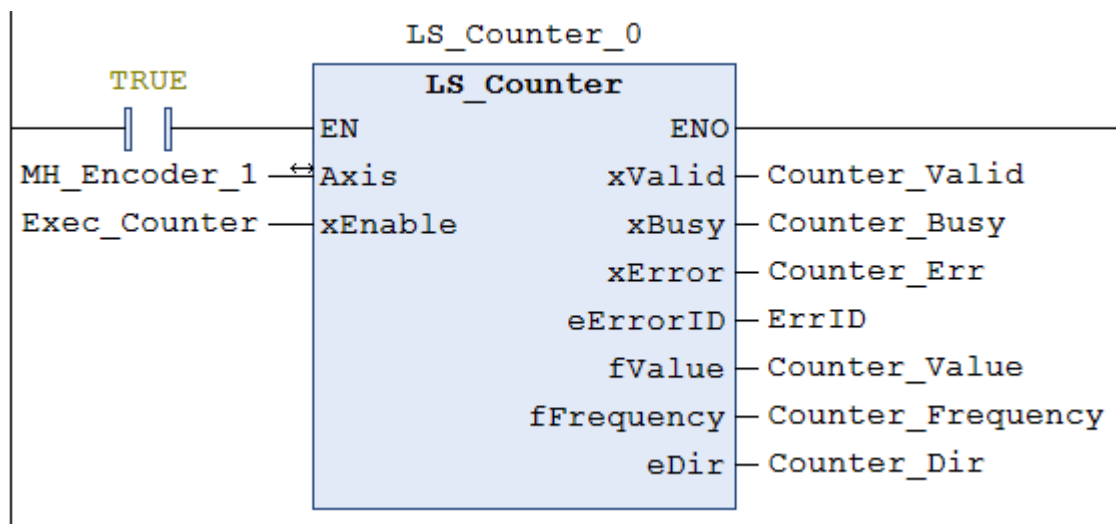
Создайте программный блок, например **POU_TouchProbe**:



и привяжите его к основной Задаче с приоритетом 0 (обработка шины обычно).



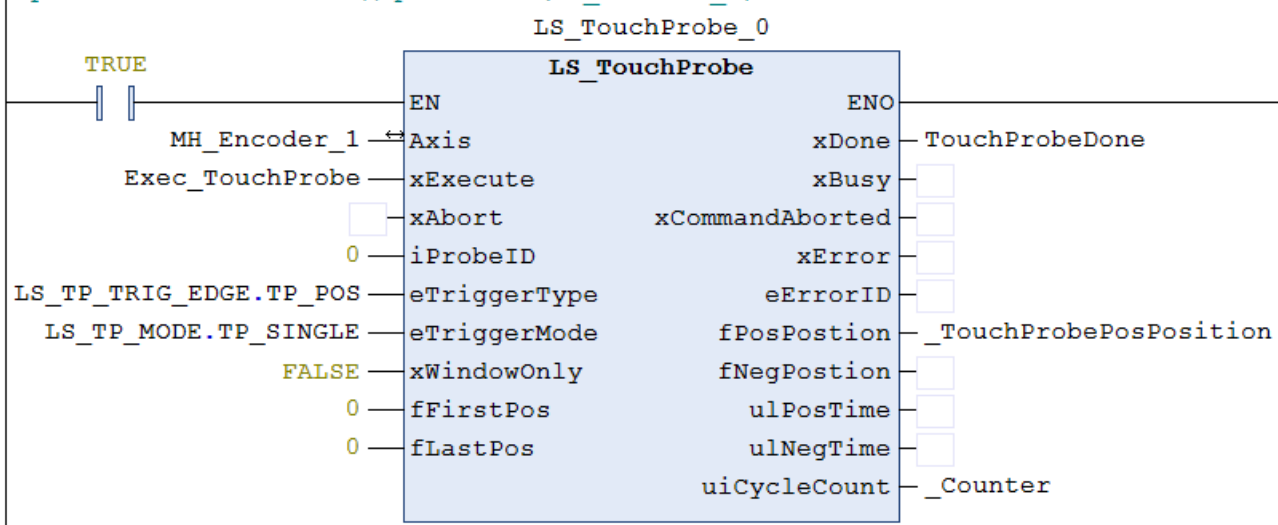
Поставьте в данный программный блок инструкцию **LS_Counter** для запуска высокоскоростного счётчика **Counter0** (обращение по имени энкодерной оси):



Поставьте в этот же программный блок инструкцию **LS_TouchProbe** для запуска функции захвата:

Запуск инструкции *Touch Probe*.

Привязка по имени энкодерной оси (*MH_Encoder_1*).



Описание основных ножек ФБ **LS_TouchProbe**:

xExecute – запуск ФБ в работу

eChannel – номер счётчика (0 или 1)

iProbeID – 0 – Probe0, 1 – Probe1

eTriggerType – TP_POS – захват в положительном направлении, TP_NEG – захват в отрицательном направлении, TP_TWO – захват в обоих направлениях

eTriggerMode – TP_SINGLE – разовый захват значения, TP_CONTINUE – непрерывный захват значения

xWindowOnly – FALSE – захват на всём диапазоне значений, TRUE – захват в разрешённом диапазоне

fFirstPos – начало разрешённого диапазона значений для захвата

fLastPos – конец разрешённого диапазона значений для захвата

xDone – работа ФБ завершена, захват выполнен

xBusy – ФБ стоит в готовности к захвату

xError – флаг ошибки

eErrorID – код ошибки

fPosPosition – захваченная позиция в положительном направлении

fNegPosition – захваченная позиция в отрицательном направлении

uiCycleCount – количество выполненных захватов

Для запуска ФБ необходимо взвести ножку **xExecute**. ФБ выставит **xBusy**. После захвата текущего значения счётчика ФБ сбросит **xBusy** и выставит флаг **xDone**. В полях **fPosPosition** или **fNegPosition** будет захваченное значение, которое будет сохраняться в данных полях до снятия сигнала **xExecute**. После снятия **xExecute** поля автоматически обнуляются.

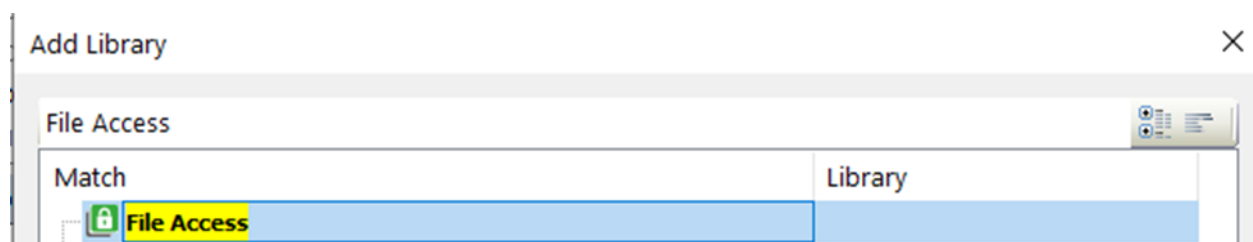
Сброс счётчика осуществляется блоком **LS_PresetValue** с нулевым смещением (**fPresetValue = 0**).

Работа с USB disc

В данной Главе рассматриваются вопрос записи файла из памяти контроллера на USB диск, вставленный в разъём USB-A на контроллере. Формат диска – FAT32. Объём до 32 Гб. Также, рассматривается обратная процедура копирования файла с USB диска в память контроллера. С целью демонстрации данного функционала будет показан механизм создания файла, но подробно операции с файлами в данной Главе не рассматриваются.

Для выполнения указанных процедур в проект должны быть добавлены следующие библиотеки:

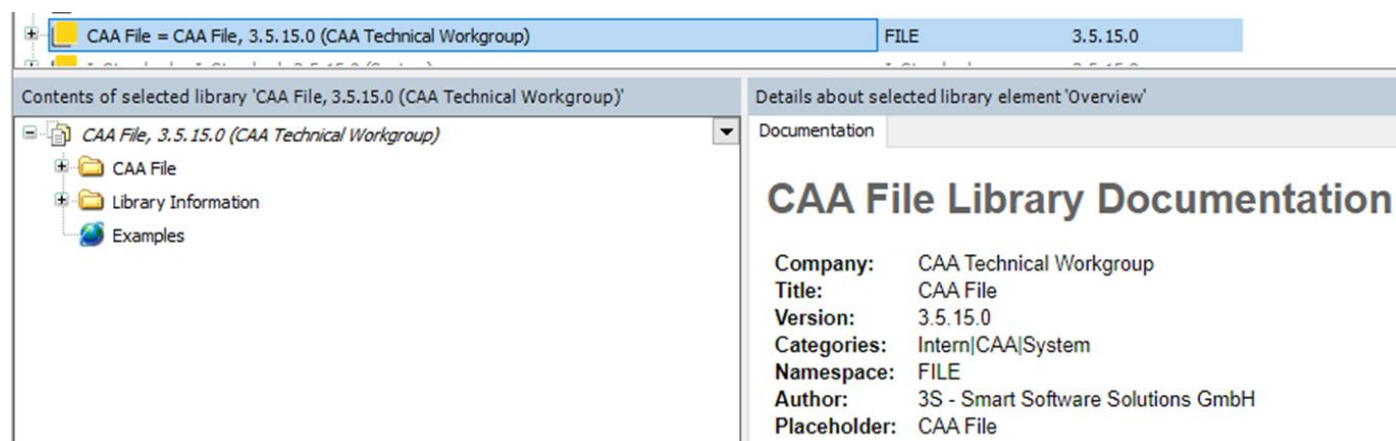
File Access



После установки в списке библиотек должны появиться два пункта **CAA File** и **CAA Types**:

File Access, 3.5.17.0 (3S - Smart Software Solutions GmbH)	File_Access	3.5.17.0
CAA File = CAA File, 3.5.15.0 (CAA Technical Workgroup)	FILE	3.5.15.0
CAA Types = CAA Types Extern, 3.5.13.0 (CAA Technical Workgroup)	CAA	3.5.13.0

Библиотека **CAA File** позволяет осуществлять операции с папками и файлами.



а **CAA Types** содержит необходимые типы данных

Contents of selected library 'CAA Types Extern, 3.5.13.0 (CAA Technical Workgroup)'	Details about selected library element 'Overview'
- CAA Types Extern, 3.5.13.0 (CAA Technical Workgroup) - CAA Types - Convert Functions - Enums - Function Blocks - GlobalConstants - GlobalVariables - Helper Functions - Types - VersionConstants - GetSupplierVersion	Documentation CAA Types Extern Library Documentation Company: CAA Technical Workgroup Title: CAA Types Extern Version: 3.5.13.0 Categories: Intern CAA Foundation Namespace: CAA Author: 3S - Smart Software Solutions GmbH Placeholder: CAA Types Description [1]

Также, потребуется ФБ **OD_FileCopy** из библиотеки **OD_Files**, которая позволяет копировать удобным способом файлы из памяти контроллера на внешние носители и обратно с возможностью создания при копировании новых папок и имени файла. Также, необходима библиотека **OD_Basic**.

OD = OD Basic, 1.0.0.1 (Optimus Drive)	OD	1.0.0.1
OD_Files = OD Files, 1.3.0.1 (Optimus Drive)	OD_Files	1.3.0.1

Contents of selected library 'OD Files, 1.3.0.1 (Optimus Drive)'	Details about selected library element 'Overview'
- OD Files, 1.3.0.1 (Optimus Drive) - DTs - OD_Files - OD_FileCopy - sOD_DirName - Reatins Objects	Inputs/Outputs Graphical ? Docu OD_FileCopy (FB) FUNCTION_BLOCK OD_FileCopy Копирование файла. В данном ФБ пр

В библиотеке **OD_Basic** содержится перечисление **FileDevice (ENUM)**

Contents of selected library 'OD Basic, 1.0.0.1 (Optimus Drive)'
- OD Basic, 1.0.0.1 (Optimus Drive) - DTs n Vars - CONSTANTS_LANGUAGE - stDT - Constants - ERROR (ENUM) - FileDevice (ENUM) - FBs - FNs - Images - Objects

позволяющее удобным способом выбрать источник и приёмник файла:

FileDevice (ENUM)



TYPE FileDevice :

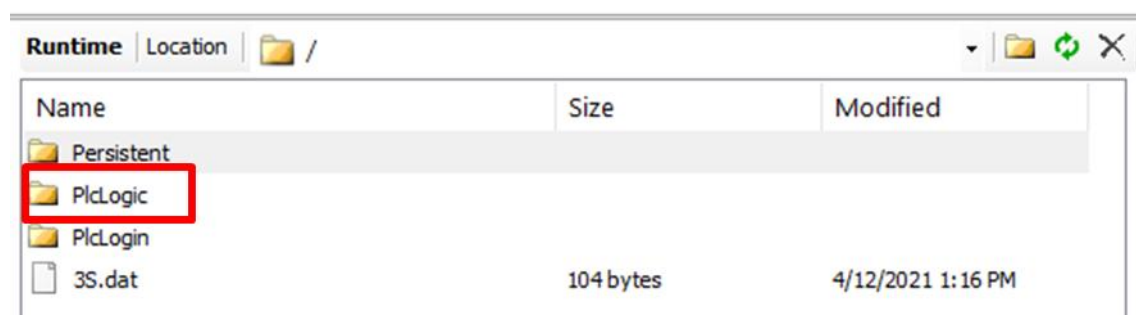
Файловое устройство

InOut:

Name	Initial	Comment
CDS_DIR	0	Рабочая директория PlcLogic CoDeSys. ВНИМАНИЕ: Для этой папки имена файлов/папок на кириллице в Files(Файлы) отображаются НЕКОРРЕКТНО!
SD	1	SD накопитель. Просим учитывать особенность MX300: Вставлять/вынимать SD карту можно только при выкл. питания контроллера!
CDS_DFD	2	Пользовательская директория CoDeSys (Data File Directory): см.Data file в IEC Controller file management
USB	3	USB накопитель. Для ME200, MX300 не поддерживает режим USB Host!
USB1	4	USB накопитель 1. Для MH1000. 1-ый - тот накопитель USB, который вставлен в USB порт (неважно в какой: слева или справа) первым!
USB2	5	USB накопитель 2. Для MH1000. 2-ой - тот накопитель USB, который вставлен в USB порт (неважно в какой: слева или справа) вторым! При обращении к внешним накопителям на SD карте/USB диске, имеющим разделы с несколькими логическими дисками, библиотека оперирует только с первым логическим диском!

В наших примерах далее будут задействованы две именованные константы из данного перечисления:

CDS_DIR – внутренняя память контроллера, папка **PlcLogic**



USB1/USB2 – вставленные в контроллер USB flash discs. Причём USB1 – это тот, который был вставлен первый, независимо от того в какой разъём, правый или левый. А USB2 соответственно тот, который был вставлен вторым.

Для процедуры копирования файлов используется ФБ **OD_FileCopy (FB)**, который берёт файл по указанному пути (папки и файл должны существовать) и помещает по указанному пути. Причём в приёмнике будут созданы указанные в пути папки, если они не существовали, а файл при копировании может быть переименован:

OD_FileCopy (FB)

FUNCTION_BLOCK OD_FileCopy

Копирование файла. В данном ФБ применяются следующие правила: 1. Если папка CDS_DFD) готово к работе, то папка(-и) будет(-ут) создана(-ы). 2. Если файл назна

InOut:

Scope	Name	Type	Initial	
Input	xExecute	BOOL		
	eFileDevSource	OD.FileDevice	OD.FileDevice.CDS_DIR	↘
	sDirPathSource	STRING(120)		↘
	sFileNameSource	STRING		↘
	eFileDevDest	OD.FileDevice	OD.FileDevice.CDS_DIR	↘
	sDirPathDest	STRING(120)		↘
	sFileNameDest	STRING		↘
Output	xDone	BOOL		
	xBusy	BOOL		
	xError	BOOL		
	eERROR	ERROR		
	dwFileSize	__XWORD		↗

Ножка ФБ	Описание
xExecute -	Выполнить ФБ
eFileDevSource	Источник, откуда копируется файл. Память контроллера или SD карта
sDirPathSource	Папка (папки) откуда копируется файл. Разделитель '/'
sFileNameSource'	Укажите имя файла в источнике с расширением. (без символов '/')
eFileDevDest	Приёмник, куда копируется файл. Память контроллера или SD карта
sDirPathDest	Папка (папки) куда копируется файл. Разделитель '/'
sFileNameDest	Укажите имя файла в приёмнике с расширением. (без символов '/')
xDone	Флаг окончания процедуры копирования файла
xBusy	Флаг работы ФБ
xError	Флаг ошибки копирования файла
eERROR	Код ошибки
dwFileSize	Размер скопированного файла в байтах

Для демонстрации процедуры доступа на USB диск для чтения-записи файлов необходимо последовательно выполнить следующие действия (в скобках указаны используемые библиотеки):

Для записи файла на USB диск:

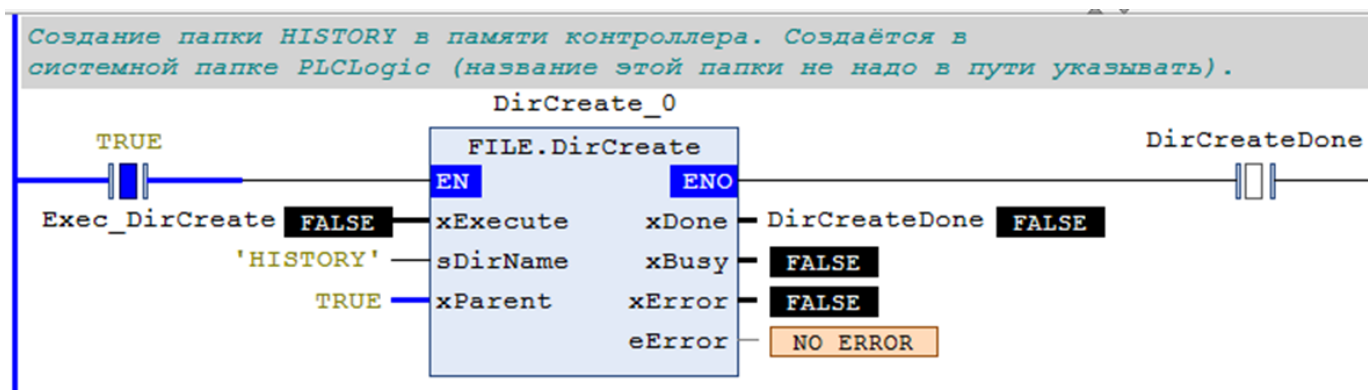
1. Создать папку в памяти контроллера в системной папке PLCLogic (CAA File)
2. Создать файл в этой папке (CAA File)
3. Открыть файл (CAA File)
4. Записать данные в файл в виде строковой переменной (программный код)
5. Закрыть файл (CAA File)
6. Скопировать файл на USB диск с созданием папки (OD_FileCopy)

Для чтения файла с USB диска:

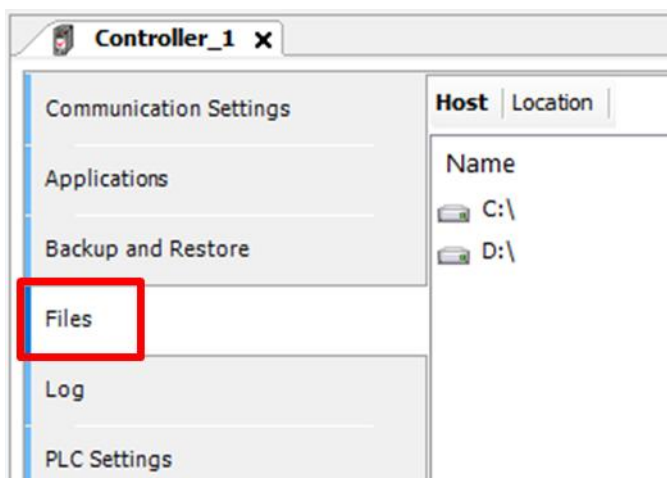
1. Создать на ПК папку и файл, записать на USB диск (пользователь на ПК)
2. Скопировать файл с USB диска в память контроллера с созданием папок (OD_FileCopy)
3. Посмотреть через среду разработки, что файл записался в память контроллера (SoftWare)

Копирование файлов из памяти контроллера на USB диске

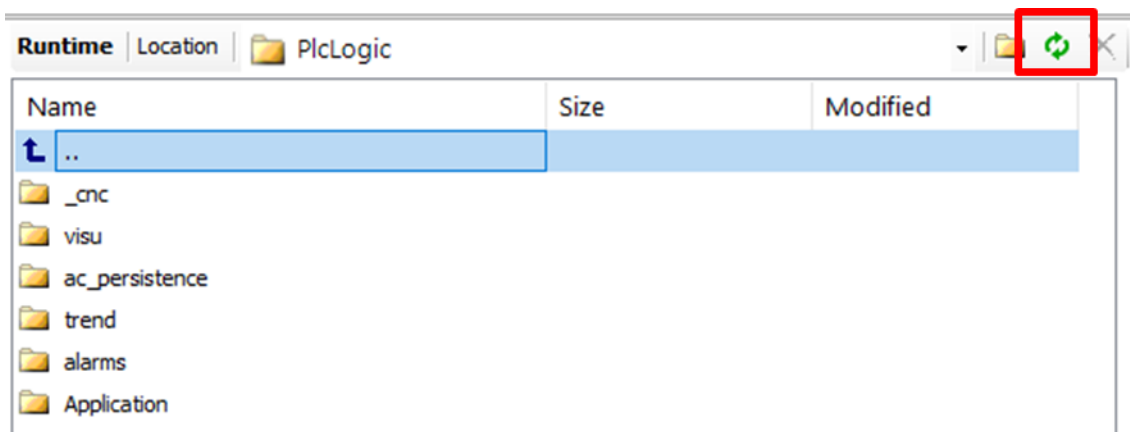
Создайте папку HISTORY в памяти контроллера при помощи ФБ **FILE.DirCreate**. Папка будет создана в системной папке **PLCLogic**. Название папки указывается в прямых кавычках (как для всех строковых переменных).



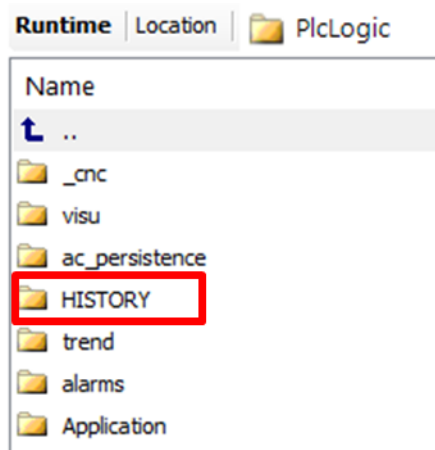
Создание папки можно проверить через среду программирования. Для этого откройте вкладку **Device** и выберите пункт **Files**:



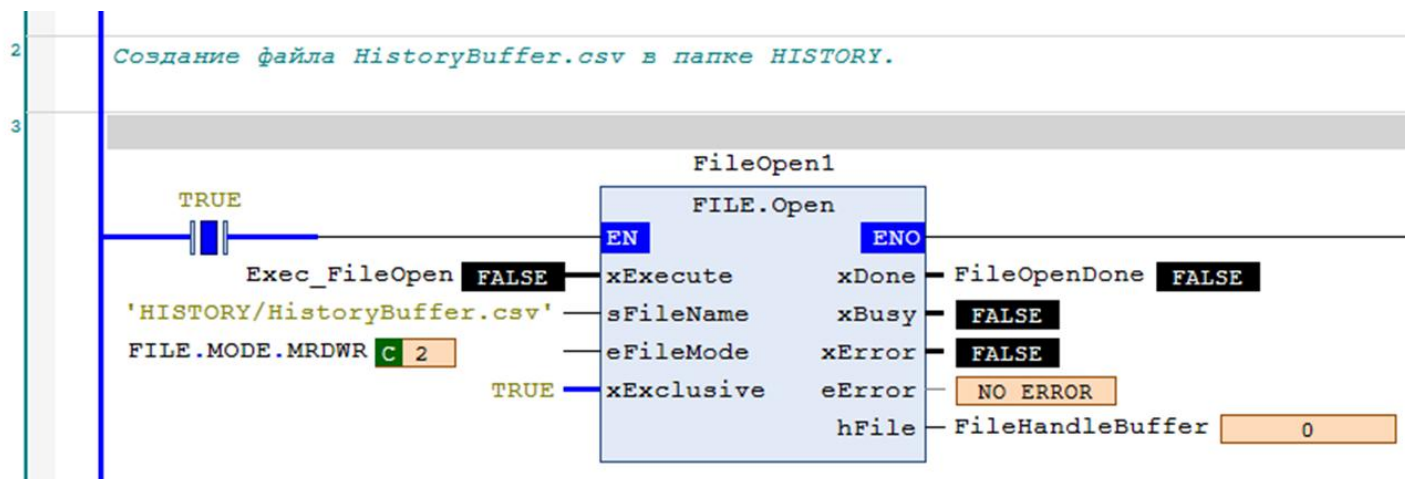
В правой части вкладки нажмите кнопку обновления информации, откройте папку **PLCLogic** и снова обновите:



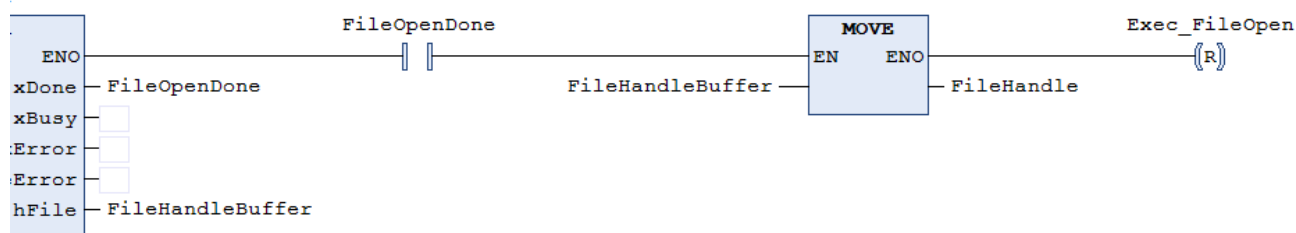
В папке PlcLogic должна быть видна созданная папка HISTORY:



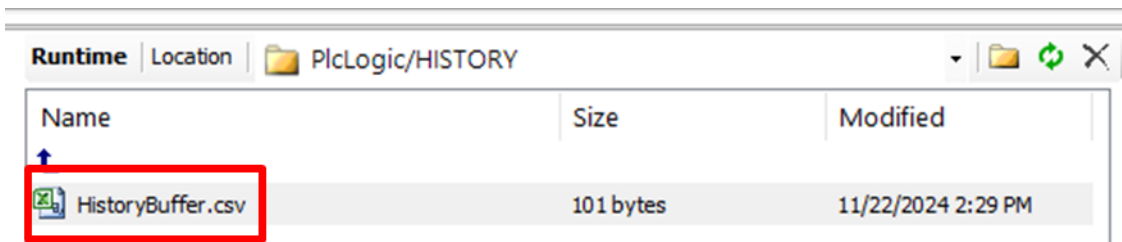
Далее создайте файл HistoryBuffer.csv в папке HISTORY при помощи ФБ **FILE.Open**.



Для сохранения дескриптора файла **hFile** по ножке **xDone** переместите его в другую переменную:



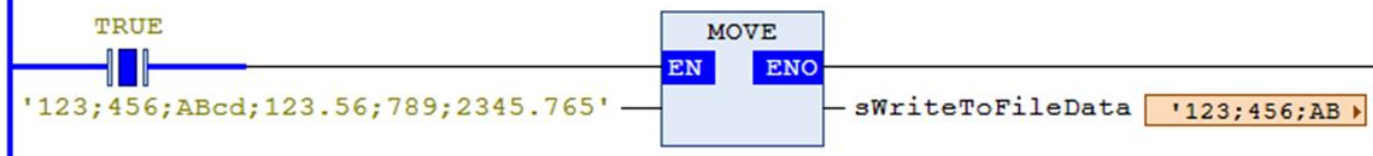
Проверить создание файла можно также в среде программирования обновив содержимое:



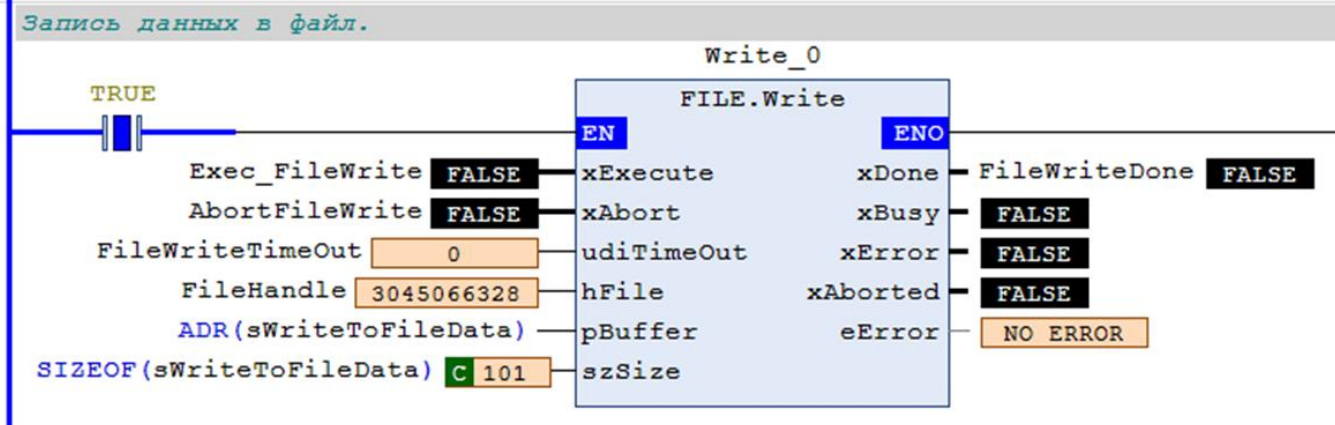
Далее запишите данные в строковую переменную, например просто скопировав строковую константу в строковую переменную.

(в данной Главе не рассматриваются операции с данными для записи в файл, поэтому используется простейший и кратчайший способ исключительно для целей демонстрации)

Данные в формате стринг, чтобы записать одной командой в файл.
(если записывать в числовом формате, то придётся к цикле по одному символу).

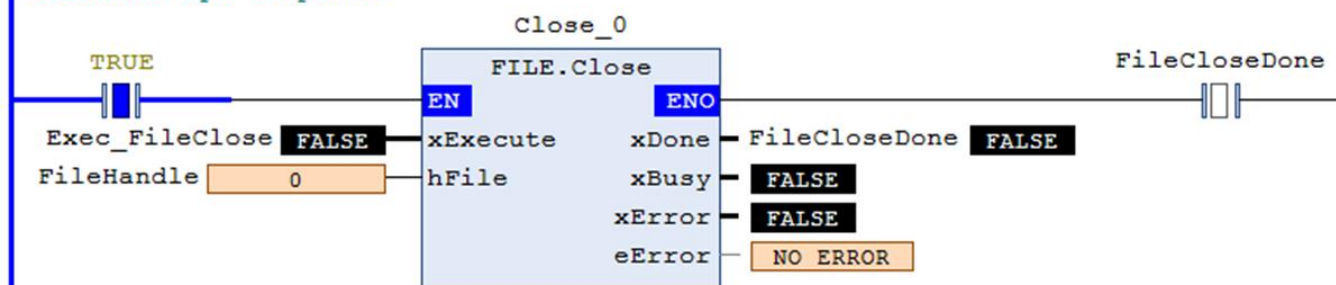


Переместите данные из переменной в файл при помощи ФБ **FILE.Write**:

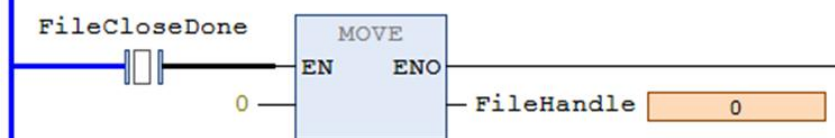


Закройте файл при помощи ФБ **FILE.Close** и обнулите переменную, хранящую дескриптор файла:

Закрытие файла HistoryBuffer.csv в папке HISTORY. Используется дескриптор файла, созданный при открытии.



Обнуление переменной с дескриптером файла.

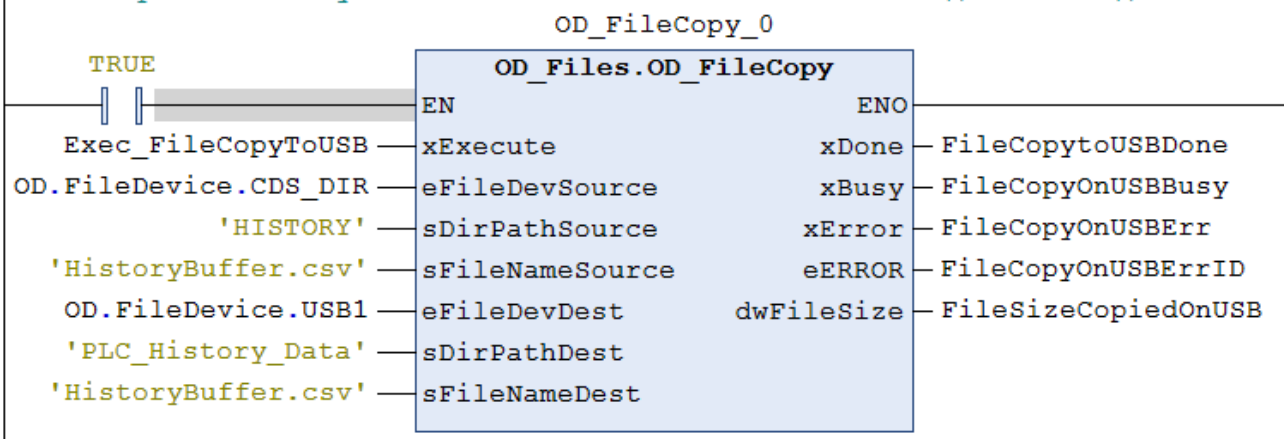


Вставьте SD карту в слот на контроллере. Карта должна иметь файловую систему FAT32 и объем не более 32 Гб.

Тип: Съемный диск
Файловая система: FAT32

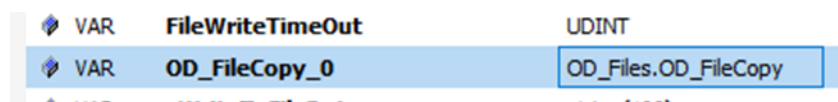
При помощи ФБ **OD_FileCopy** скопируйте файл HistoryBuffer.csv из папки HISTORY на диск USB1 с созданием папки PLC_History_Data. Название файла не меняется, но при необходимости при копировании название файла можно задать новое. Содержимое при этом останется как у исходного файла.

Запись файла HistoryBuffer.csv в папке HISTORY на USB1 диск с созданием папки



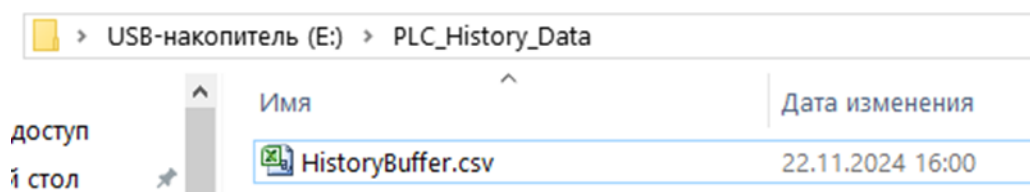
На ножке **dwFileSize** будет отображаться размер записанного файла в байтах.

При объявлении экземпляра ФБ **OD_FileCopy** и фактических переменных для него необходимо использовать при определении типа переменной имя библиотеки **OD_Files**:



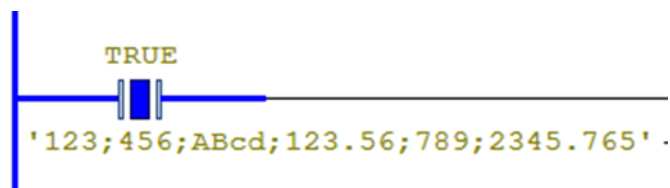
23	VAR	FileCopyOnSDErr	BOOL
24	VAR	FileCopyOnSDErrID	OD_Files.ERROR

Для проверки записи файла извлеките USB диск из контроллера и вставьте в ПК. В проводнике можно будет увидеть в корне диска папку и в ней файл:



Запустите файл через Excel для отображения содержимого файла, которое будет соответствовать записанным в программе данным.

Программа:



Содержимое файла:

Буфер обмена

ГМ

Шрифт

ГМ

Выравни

B14

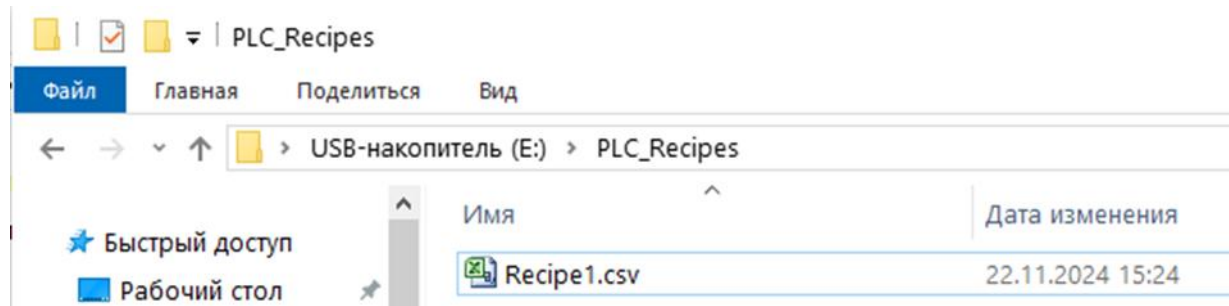
fx

	A	B	C	D	E	F
1	123	456	ABcd	123.56	789	2345.765
2						

Копирование файлов с SD карты в память контроллера

Вставьте в ПК USB диск. Формат файловой системы должен быть FAT32 и объем не более 32 Гб.

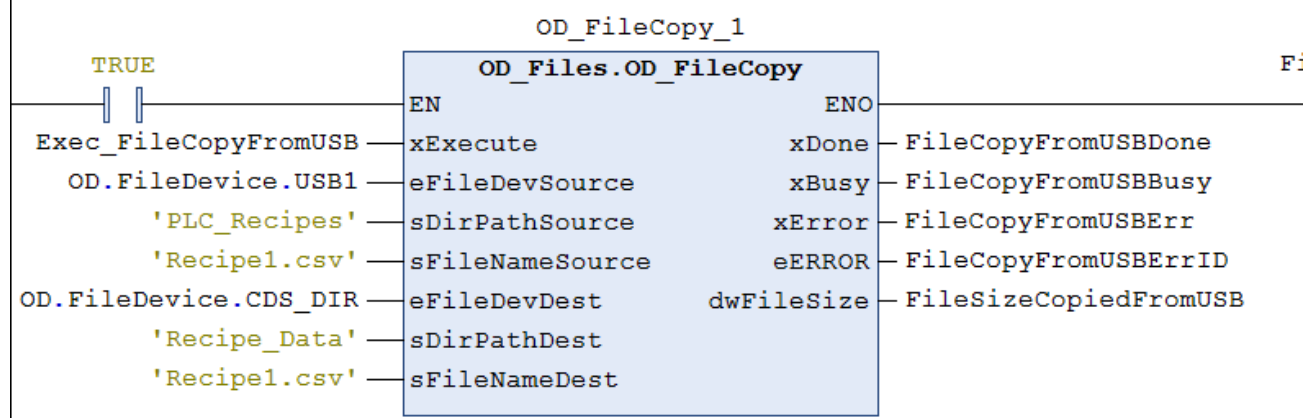
Запишите на диск средствами ПК папку PLC_Recipe и файл Recipe1.csv:



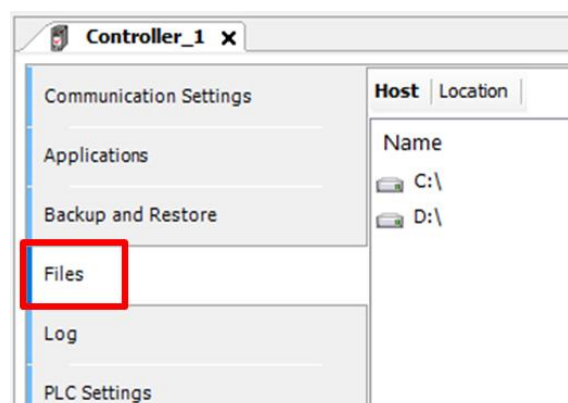
Вставьте карточку в слот на контроллере.

При помощи ФБ **OD_FileCopy** скопируйте папку PLC_Recipe и файл Recipe1.csv во внутреннюю память контроллера (папка **PLCLogic**) с созданием папки Recipe_Data.

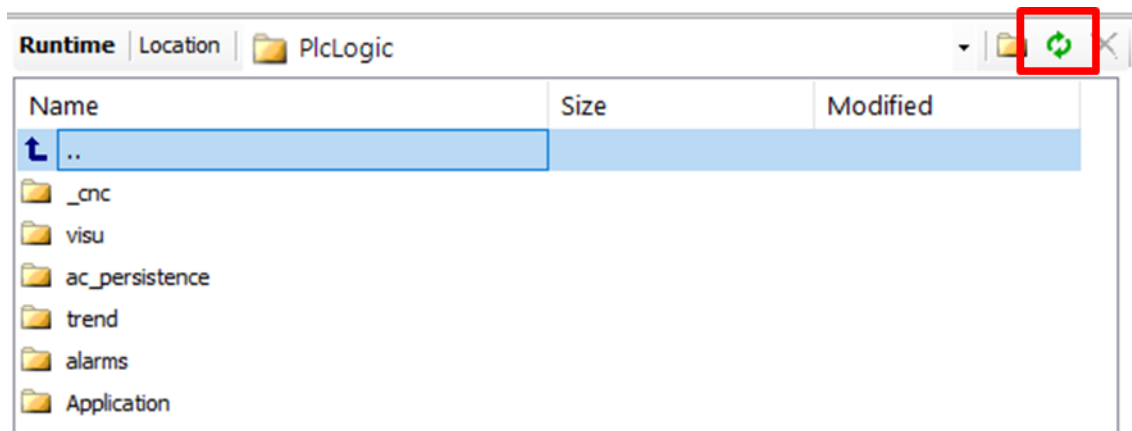
Копирование файла Recipe1.csv из папки PLC_Recipes с USB1 диска во внутреннюю память контроллера (папка PLCLogic с созданием папки Recipe_Data. (не забудьте вставить USB диск в контроллер)



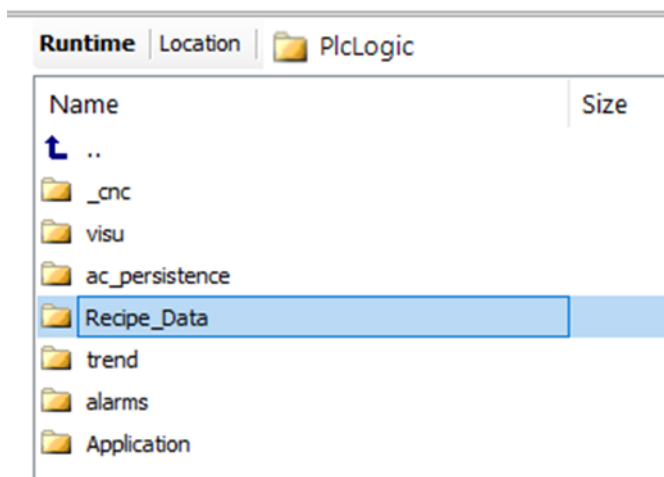
Создание папки можно проверить через среду программирования. Для этого откройте вкладку **Device** и выберите пункт **Files**:



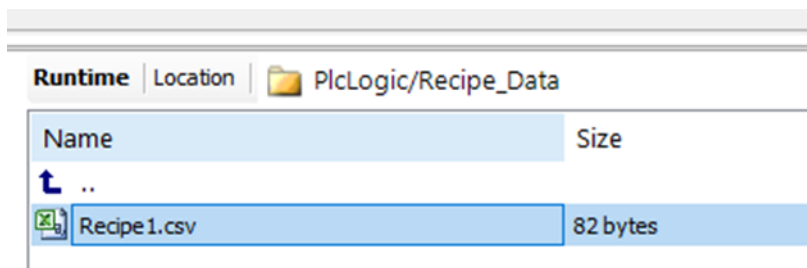
В правой части вкладки нажмите кнопку обновления информации, откройте папку **PLcLogic** и снова обновите:



В папке PLcLogic должна быть видна созданная папка Recipe_Data:



и в ней файл Recipe1.csv:



Работа в режиме ШИМ импульсов

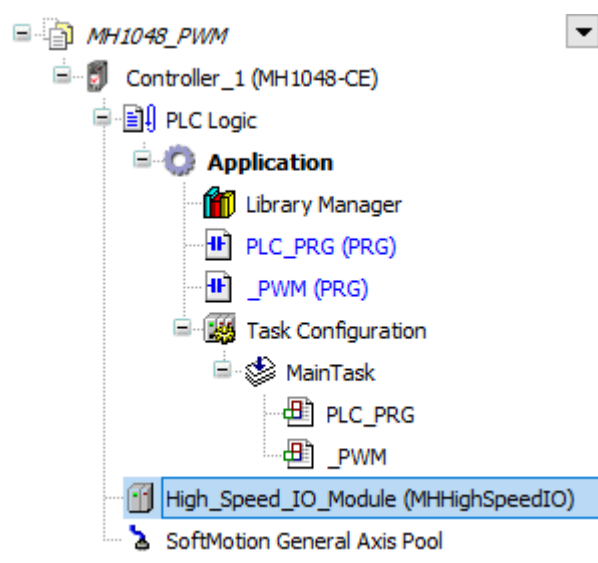
Контроллеры типа MH1048 имеют аппаратную возможность формировать ШИМ импульсы на четырёх выходах 4, 5, 6 и 7. В проект должна быть добавлена библиотека **LS_HSIO**:

LS_HSIO = LS_HSIO, 1.0.3.7 (Leadshine Technology Co.Ltd)

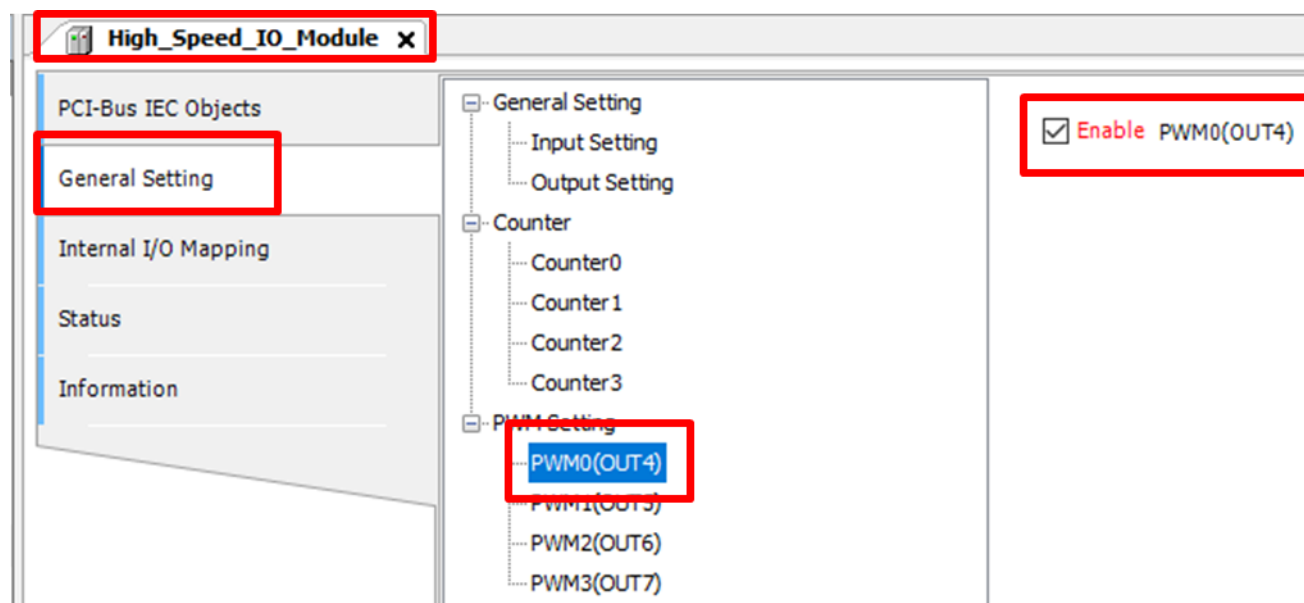
LS_HSIO

1.0.3.7

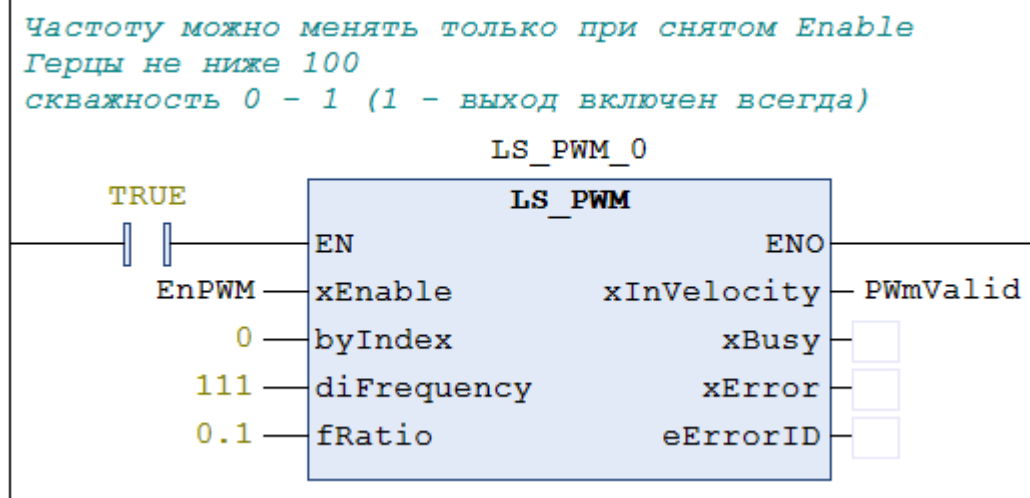
Для организации данного функционала в проекте должен быть адаптер входов-выходов **MHHighSpeedIO** (см. соответствующий раздел данного Руководства):



Щёлкните дважды левой кнопкой мышки на пункте **MHHighSpeedIO** и в открывшейся вкладке выберите пункт **General Configuration – PWM Setting – PWM0(OUT4)**. Поставьте флажок **Enable PWM0(OUT4)**:



Создайте программный блок и привяжите его в **Task Configuration** к основной задаче проекта (приоритет 0). Обычно это **EtherCAT Task**. Добавьте в программу команду **LS_PWM**:



Описание ножек ФБ:

xEnable – запуск ФБ. ШИМ выдаётся пока есть TRUE

byIndex – номер выхода, 0 – OUT4, 1 – OUT5, 2 – OUT6, 3 – OUT7

diFrequency – частота ШИМ импульсов. Диапазон 100 Гц – 200 кГц

fRatio – скважность 0 -1 REAL. 0 – выход полностью закрыт, 1 – выход полностью открыт

xInVelocity – флаг, что импульсы выдаются

xBusy – флаг, что ФБ в работе

xError – флаг ошибки

eErrorID – код ошибки