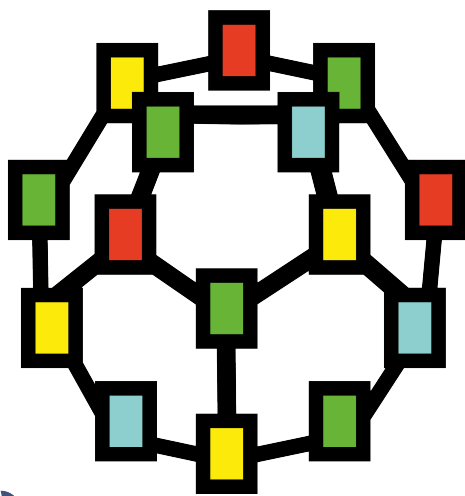




Библиотека raSync



Руководство пользователя

09.2024
версия 1.1

Содержание

Используемые термины и сокращения	3
Введение	4
1 Общие сведения об организации резервирования в ОВЕН ПЛК210	5
1.1 Аппаратное резервирование питания контроллера.....	5
1.2 Программное резервирование контроллеров	5
2 Библиотека raSync	8
2.1 Менеджер синхронизации (SyncMan).....	9
2.2 Определение ведущего (MasterSel)	11
3 Создание синхронизированных модулей	14
4 Синхронизация модулей. Менеджер синхронизации (SyncMan)	16
5 Определение ведущего контроллера (MasterSel)	22
5.1 Пример реализации резервирования ПЛК с общими Mx210.....	22
5.2 Пример реализации резервирования ПЛК с индивидуальными Mx210.....	32

Используемые термины и сокращения

Ведомый контроллер – контроллер, который находится в «горячем» резерве и синхронизирует данные с ведущим контроллером.

Ведущий контроллер – контроллер, с которым синхронизируется ведомый контроллер. В зависимости от реализованной схемы резервирования ведущий контроллер может принимать на себя роль мастера для линейки модулей ввода/вывода, выдавать сигналы на каналы вывода на своем борту и т.д.

Определение ведущего контроллера или переключение роли ведущего между контроллерами – алгоритм выбора текущей роли контроллера: ведущий или ведомый. Может быть как автоматическим, так и ручным (по команде оператора).

ОС – операционная система.

ПКМ – правая кнопка мыши.

ПЛК – программируемый логический контроллер.

Резервирование (по ГОСТ 27.002-89) – способ обеспечения надежности объекта за счет использования дополнительных средств и (или) возможностей, избыточных по отношению к минимально необходимым для выполнения требуемых функций.

Резервирование замещением (по ГОСТ 27.002-89) или **100 % «горячее» резервирование** – резервирование, при котором функции основного элемента передаются резервному только после отказа основного элемента.

Синхронизация данных – устранение различий между двумя аналогичными наборами данных контроллеров посредством обмена информацией по выделенным линиям связи.

Runtime – вычислительное окружение, необходимое для выполнения компьютерной программы и доступное во время выполнения компьютерной программы.

SQL (Structured Query Language) – язык программирования для хранения и обработки информации в реляционной базе данных.

Введение

Настоящее руководство описывает синхронизацию проектов и организацию резервирования для контроллеров ОВЕН, программируемых в среде Полигон. Предполагается, что читатель обладает базовыми навыками работы с Полигон, поэтому общие вопросы (например, создание и загрузка проектов) в данном документе не рассматриваются – они подробно описаны в документах [Руководство по программированию](#), [Библиотека raCore](#) и [Быстрый старт](#).

Синхронизация проектов и организация резервирования в среде Полигон осуществляется с помощью функциональных блоков из библиотеки **paSync**. Данная библиотека доступна для работы при наличии соответствующей лицензии runtime (см. описание лицензионных пакетов [на странице среды разработки Полигон](#)).

Документ соответствует версии среды Полигон – **1994**, версии библиотеки **paSync** – **58** и выше.

1 Общие сведения об организации резервирования в ОВЕН ПЛК210

ОВЕН ПЛК210 с исполнительной средой Полигон поддерживают:

- Горячее аппаратное резервирование питания контроллера – см. [раздел 1.1](#);
- Горячее программное резервирование приложения пользователя – см. [раздел 1.2](#).

1.1 Аппаратное резервирование питания контроллера

В контроллерах ПЛК210 предусмотрено два порта для подключения источников питания **24 В**:


- **Порт 1** – основное питание;
- **Порт 2** – резервное питание.

Переход на резервное питание происходит при снижении напряжения основного питания менее **9 В**.

При восстановлении работоспособности основного источника питания контроллер автоматически возвращается на питание от основного источника.



ПРИМЕЧАНИЕ

Информацию о наличии питающего напряжения на портах контроллера можно получить с помощью блока **210-Power** из библиотеки **paOwenIO**. Также блок **210-Power** позволяет задать режим индикации светодиода **Батарея** . Подробнее см. в документе [Работа с ОВЕН ПЛК. Библиотека paOwenIO](#).

1.2 Программное резервирование контроллеров

Программное резервирование пользовательского приложения Полигон реализуется с помощью блоков из библиотеки **paSync** (см. описание библиотеки в [разделе 2](#)).

Библиотека **paSync** доступна для постоянной работы при наличии соответствующей лицензии runtime контроллера (см. описание лицензионных пакетов [на странице среды разработки Полигон](#)).

Среда разработки **Полигон** предоставляет следующий функционал при организации резервирования контроллеров:

1. Дублирование (полное или частичное) пользовательских программ (модулей контроллеров в проекте Полигон) на стадии разработки.
2. Синхронизация данных дублированных программ контроллеров во время исполнения – блоков ***_sync** и данных в разделе блока **SyncMan** из библиотеки **paSync**.
3. Автоматическое переключение ролей контроллеров ведущий и ведомый – блок **MasterSel** из библиотеки **paSync**.
4. Ручное переключение ролей контроллеров ведущий и ведомый – блок **MasterSel** из библиотеки **paSync**.
5. Среда не ограничивает пользователя в создании собственного алгоритма переключения ролей ведущий и ведомый контроллеров.

Варианты схем резервирования ОВЕН ПЛК с исполнительной средой **Полигон** практически не ограничены и могут модернизироваться в соответствии с требованиями конкретного автоматизируемого технологического объекта.

Примеры схем организации резервирования ОВЕН ПЛК с исполнительной средой **Полигон**:

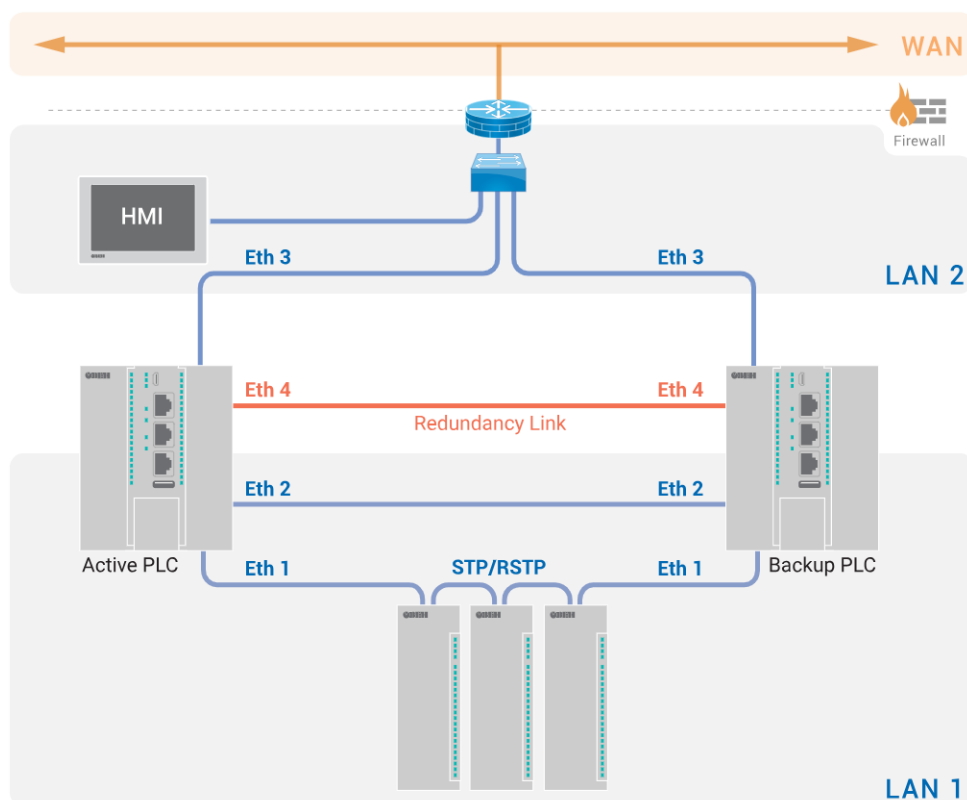


Рисунок 1.1 – Схема резервирования контроллеров с общей линейкой модулей Mx210

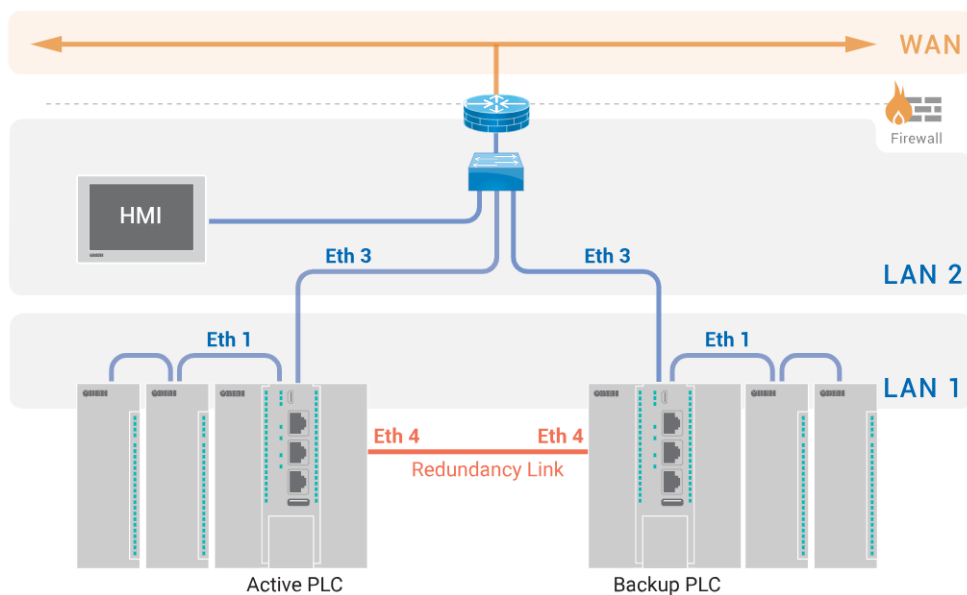


Рисунок 1.2 – Схема резервирования контроллеров с индивидуальными линиями модулей Mx210

**ПРИМЕЧАНИЕ**

Модификации резервированной пары контроллеров не обязательно должны совпадать.

**ПРИМЕЧАНИЕ**

Настройка режимов работы сетевых интерфейсов контроллера производится в web-интерфейсе конфигурации (см. [Руководство по эксплуатации](#)).

На этапе разработки проекта осуществляется полное или частичное дублирование пользовательских программ контроллеров (см. пример настройки в [разделе 3](#)).

После запуска дублированных программ на контроллерах следует определение ведущего и ведомого контроллера (ручное или автоматическое) с помощью готового алгоритма, реализуемого блоком **MasterSel**, или алгоритмом пользователя.

Оба контроллера циклически выполняют пользовательское приложение.

Ведомый контроллер начинает работать в режиме **OPC UA**-клиента (блок **SyncMan**) и оформляет подписку на данные (блоки *_**sync** и раздел блока **SyncMan**) **OPC UA**-сервера ведущего контроллера. Таким образом, ведомый контроллер синхронизирует свои данные с данными ведущего контроллера (см. пример в [разделе 4](#)).

Обмен диагностическими сигналами двух контроллеров можно организовать по двум изолированным интерфейсам **Ethernet** с помощью готового блока **MasterSel** (см. примеры настройки в [разделе 5](#)).

Условия переключения ведущего контроллера при использовании блока **MasterSel** описаны в справке среды Полигон и в [разделе 2.2](#).

Алгоритм работы резервированной пары контроллеров при использовании блока **MasterSel**:

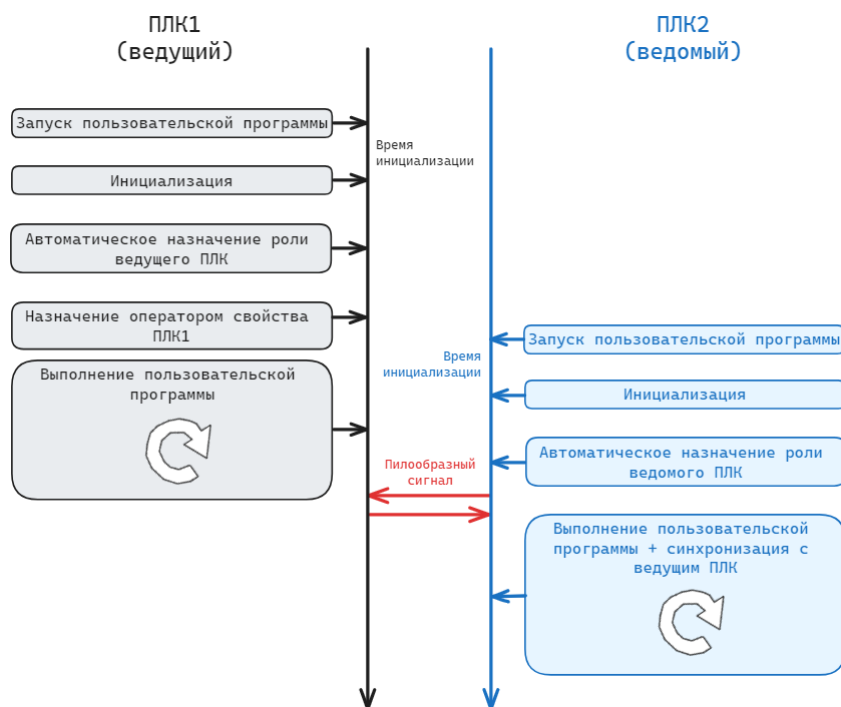


Рисунок 1.3 – Алгоритм работы резервированной пары контроллеров при использовании блока **MasterSel**

Примеры настройки резервированной пары контроллеров при использовании блока **MasterSel** приведены в [разделе 5](#).

2 Библиотека raSync

raSync – библиотека, обеспечивающая синхронизацию данных между дублированными модулями контроллеров и организацию резервирования.

Один из контроллеров выбирается **ведущим** – с ним синхронизируется **ведомый** контроллер. Синхронизация сигналов во время работы контроллеров обеспечивается для блоков с поддержкой синхронизации из библиотеки **raSync**, которые реализуют базовые алгоритмы аналогично блокам из библиотек **raCore** и **raControls**. Работа данных блоков описана в справке среды и в этом документе не рассматривается.

Для добавления библиотеки **raSync** в проект следует:

1. Перейти в меню **Окна/Проекты**. В появившемся окне отобразится текущий проект и добавленные библиотеки.

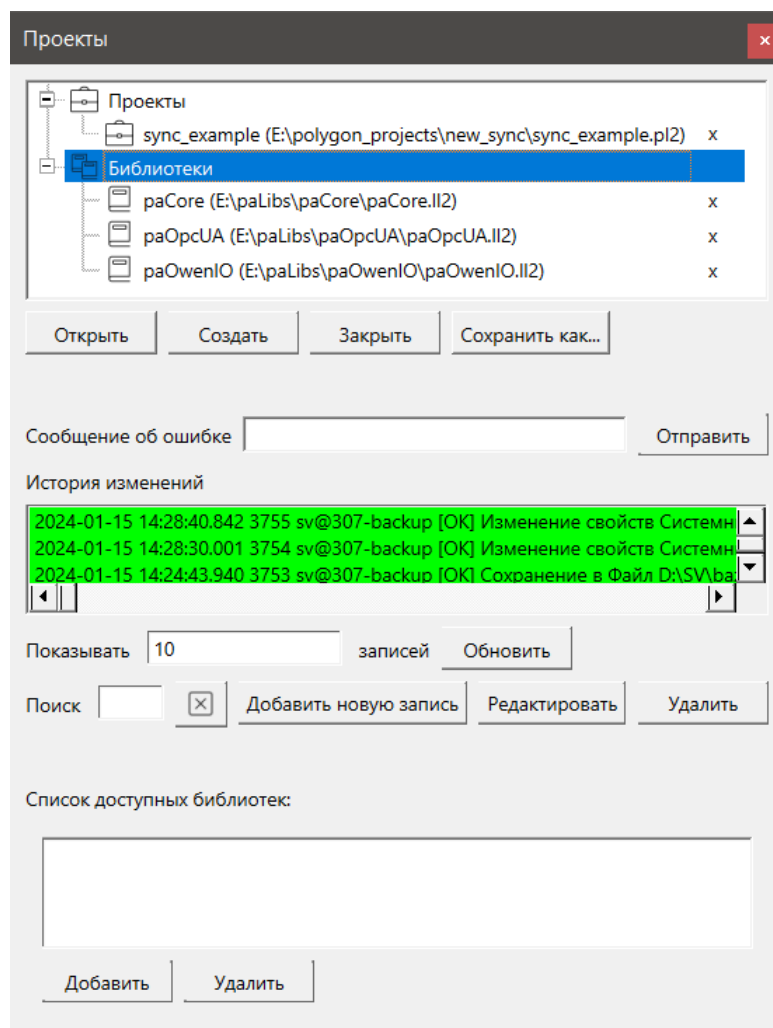


Рисунок 2.1 – Добавление библиотеки raSync в проект

2. Нажать кнопку **Открыть** и перейти в папку с файлами библиотеки, которую необходимо добавить.
3. В выпадающем списке выбрать тип файла **Библиотека Полигон 2 (*.II2)**.

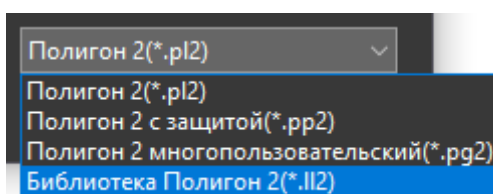


Рисунок 2.2 – Добавление библиотеки raSync в проект

4. В окне появится файл библиотеки с расширением **.II2**. Следует выбрать его и нажать **Открыть**.

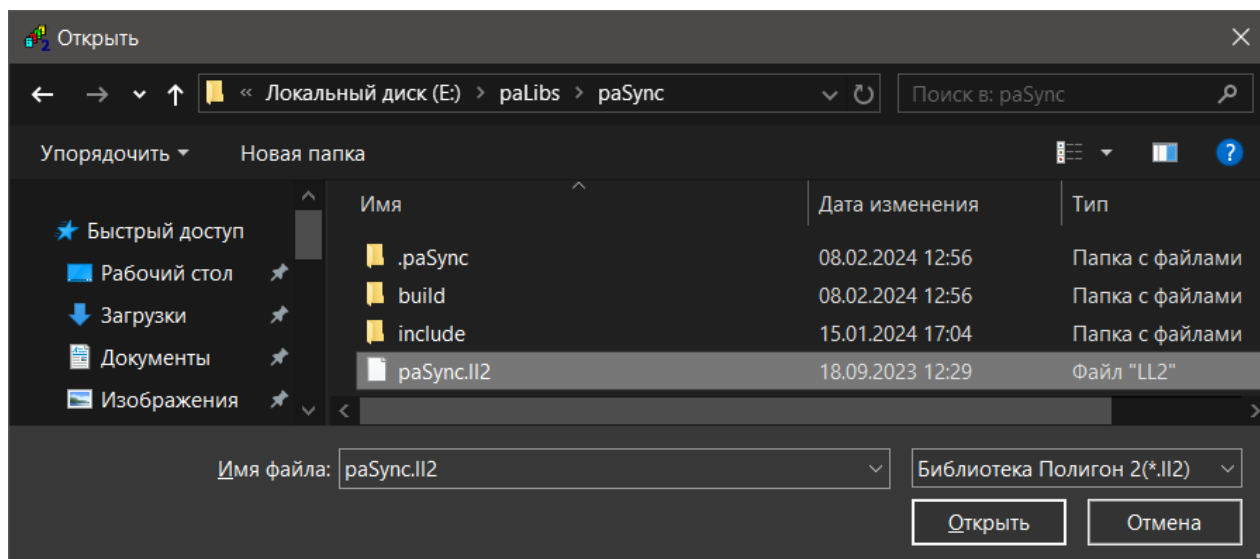


Рисунок 2.3 – Добавление библиотеки paSync в проект

Добавленная библиотека отобразится в окне *Проекты*.

2.1 Менеджер синхронизации (SyncMan)

Блок **SyncMan** обеспечивает синхронизацию данных между **ведущим** и **ведомым** контроллерами. Обмен реализован через протокол **OPC UA**, в ведомом контроллере создается подписка на изменение данных от ведущего.

SyncMan основан на блоке **OpcUAClient** из библиотеки **paOpcUA**. Подробнее реализация протокола **OPC UA** в среде Полигон описана в документе [Обмен с верхним уровнем. Библиотека paOpcUA](#).

Данный блок можно разместить только в **Фоне**.


Данные, необходимые для синхронизации блоков *_sync библиотеки **paSync**, добавляются в подписку автоматически. Дополнительные входы/выходы, которыми необходимо обмениваться с сервером, должны быть добавлены в раздел **Данные** внутри этого блока.




ПРИМЕЧАНИЕ

Если блоки библиотеки **paSync** находятся внутри составных блоков, то допускается до 7 вложенностей.

Таблица 2.1 – Назначение входов и выходов SyncMan

Элемент	Описание
Входы	
enb	Разрешение на работу блока
sync	Синхронизация: 0 – выключена; 1 – включена  ПРИМЕЧАНИЕ На данный вход может быть подан инвертированный сигнал с выхода L_Master блока MasterSel или пользовательского блока, реализующего алгоритм определения ведущего контроллера.
wait	Таймаут ожидания ответа от сервера, мс (константный)
lip	Локальный IP адрес (константный)
lprt	Локальный порт (константный)
sdr	Сетевой стек, для ПЛК ОВЕН "I" (константный)
rip	IP адрес сервера (константный)
rprt	Порт сервера (константный)
usr	Логин для доступа к серверу (константный)
psw	Пароль для доступа к серверу (константный)

Продолжение таблицы 2.1

Элемент	Описание
prio	Приоритет дополнительного потока (константный), в котором выполняется синхронизация, обычно устанавливается выше других фоновых потоков, чтобы обеспечить максимальную скорость синхронизации: 0 – отключает создание дополнительного потока (обмен идет в текущем фоновом потоке); 1...47 – приоритет потока  ПРИМЕЧАНИЕ Максимально возможное значение приоритета для конкретной ОС контроллера можно определить с помощью блока <i>ThreadMan</i> из библиотеки <i>paCore</i> , раздел <i>Системные</i> .
rst	Сброс максимальных значений временных счетчиков: выходов mwrk и mscan
fnum	Максимальное количество подменных во время отладки входов/выходов, которое можно синхронизировать: 0 – отключает синхронизацию подменных значений
m_rbufs	Вход для подключения блоков типа <i>BufSupEx</i> (циклический)
Выходы	
sts	Статус работы: 0 – нет обмена; 1 – обмен; 2 – в резерве; >2 – переходное состояние
svld	Синхронизация работает
sst	Статус сервера в соответствии со спецификацией OPC UA (см. Part 5 – 12.6 ServerState)
ssl	Дополнительный статус сервера ServiceLevel в соответствии со спецификацией OPC UA (см. Part 4 – 6.6.2.4.2 ServiceLevel)
sid	ID подписки
ssn	Номер уведомления подписки
rcnt	Количество принятых пакетов
wcnt	Количество отправленных пакетов
prio	Приоритет дополнительного потока (0 – отключен)
dsz	Количество синхронизируемых данных (входов и выходов блоков <i>*_sync</i> и добавленных в раздел <i>SyncMan</i>)
wrk	Текущее время работы одного цикла потока синхронизации, мс
mwrk	Максимальное время работы одного цикла потока синхронизации, мс
scan	Текущее время получения всех изменившихся данных (входов и выходов) от соседнего контроллера, мс
mscan	Максимальное время получения всех изменившихся данных (входов и выходов) от соседнего контроллера, мс
diag	Диагностический счетчик (используется для контроля получения всех данных и расчета времени)
fnumo	Количество синхронизируемых подменных во время отладки входов/выходов

	b36	15		
	SyncMan			
связь: 0 - выкл, 1 - вкл	1	enb	u8 i32 sts	0 - нет обмена, 1 - обмен, 2 - в резерве, >2 - переходное состояние, <0 - системная ошибка
синхронизация: 0 - выкл, 1 - вкл	0	sync	u8 u8 svld	синхронизация работает
таймаут (мс)	500	wait	ul32 i32 sst	статус сервера
локальный IP адрес "0.0.0.0"		lip	str u8 ssl	service level сервера
локальный порт 8000		lpri	i16 u32 sid	ID подписки
выбор сетевой платы / или /sock2 "/"		sdr	str u32 ssn	номер уведомления подписки
IP адрес сервера "0.0.0.0"		rip	str u32 rcnt	принято
порт сервера 4840		rpri	i16 u32 wcnt	отправлено
логин ""		usr	str u8 prio	приоритет дополнительного потока (0 - отключен)
пароль ""		psw	str u32 dsz	количество данных
приоритет потока (1-200), 0 - отключить	30	prio	u8 fit wrk	текущее время работы (мс)
сброс временных счетчиков	0	rst	u8 fit mwrk	максимальное время работы (мс)
синхронизировать отладчик (кол-во данных)	100	fnum	u16 fit scan	текущее время получения данных (мс)
?		buf0	bsup # fit mscan	максимальное время получения данных (мс)
			u16 diag	диагностический счетчик
			u16 fnumo	синхронизировано данных отладчика

Рисунок 2.4 – Менеджер синхронизации дублированного контроллера (SyncMan)

Дополнительные входы и выходы, которыми необходимо обмениваться с сервером, могут быть добавлены в раздел **Данные** блока **SyncMan**.

Выходы функциональных блоков, добавленные в раздел **Данные** передаются на сервер всегда в соответствии со свойствами (подробнее см. в описании блока **OpсUAClient** из библиотеки **paOpсUA**).

Входы функциональных блоков, добавленные в раздел **Данные**, читаются из сервера в соответствии со следующими правилами:

1. Если у входа есть свойство **ID источника/приемника**, то он читается из сервера всегда при наличии связи. Такие данные используются для двустороннего обмена между ведущим и ведомым независимо от текущей роли контроллера.
2. Если у входа нет свойства **ID источника/приемника**, то он читается из сервера только при включенной синхронизации (**sync = 1**). Такие данные используются для синхронизации вручную.

Для синхронизации параметров можно использовать блоки **BufSupEx**, подключенные к входам **m_rbufs** блока **SyncMan** (дополнительные входы добавляются командой **Создать**). Значения параметров блока **BufSupEx** синхронизируются только при **sync = 1**.

Пример работы с блоком приведен в [разделе 2.1](#).

2.2 Определение ведущего (MasterSel)

Блок **MasterSel** определяет роли **ведущий/ведомый** для двух контроллеров в резервированной конфигурации.

Таблица 2.2 – Назначение входов и выходов MasterSel

Элемент	Описание
Входы	
me1	Признак ПЛК1 (me1 = 1 у ПЛК1). Под ПЛК1 подразумевается тот контроллер, который должен становится ведущим при неопределенных условиях (восстановление связи между двумя работающими контроллерами)
ready	Внешнее условие готовности контроллера (разрешение стать основным)
init	Запуск таймера на инициализацию
master1	Команда ПЛК1 стать ведущим (внутри выделяется фронт с 0 на 1)
master2	Команда ПЛК2 стать ведущим (внутри выделяется фронт с 0 на 1)
tpila	Таймер залипания пилы, мс
trecon	Таймер восстановления связи, мс
tinit	Таймер на инициализацию, мс
Выходы	
L_Master	Признак ведущего: 0 – данный контроллер ведомый; 1 – данный контроллер ведущий
L_pila	Диагностика: генерируемый пилообразный сигнал этого контроллера
L_ready	Диагностика: готовность этого контроллера

Продолжение таблицы 2.2

Элемент	Описание
L_init_over	Инициализация завершена
conn_fault	Нет связи с соседним контроллером
R_pila_1	Диагностика: пила соседнего контроллера по каналу связи 1
R_pila_2	Диагностика: пила соседнего контроллера по каналу связи 2
R_ready	Диагностика: готовность соседнего контроллера
R_master	Диагностика: соседний контроллер ведущий

После запуска программы контроллера, выход блока **L_Master = 0**, т.е. ПЛК является ведомым.

Срабатывание любого из условий переключения в роль ведущего возможно только после завершения инициализации (выход **L_init_over = 1**). Инициализация считается завершённой после окончания отсчета времени **tinit** от появления единицы на входе **L_init** (обычно сразу установлен в 1).

Контроллер может стать **ведущим** в случаях:

- Если он ведомый и готов (**L_Master = 0** и **L_ready = 1**), а соседний – не готов (**R_ready = 0**);
- По получению соседним ведущим контроллером команды на смену мастера (**master1** или **master2**). Тогда этот контроллер станет ведущим, если он готов, или оба контроллера не готовы;
- После восстановления связи между ПЛК, когда оба ведомые, и этот контроллер – **ПЛК1** (вход **me1 = 1**);
- При потере связи с соседним ПЛК (**conn_fault = 0**).

Контроллер может стать **ведомым** в случаях:

- Если он ведущий и не готов (**L_Master = 1** и **L_ready = 0**), а соседний – готов (**R_ready = 1**);
- По получению им команды на смену мастера (**master1** или **master2**). Тогда другой контроллер станет ведущим, если он готов, или оба контроллера не готовы;
- После восстановления связи между контроллерами, когда оба ведущие, и этот ПЛК – **ПЛК2** (вход **me1 = 0**).

В иных случаях смены ролей не происходит.

Готовность контроллера (выход **L_ready** блока) формируется на основании двух условий по логике «И»:

- Есть внешнее условие готовности, заводимое на вход **ready**;
- Инициализация связи завершена.

С соседним контроллером через две выделенные линии синхронизации производится обмен данными и контролируется связь посредством передачи по каждой линии пилообразного сигнала:

- Если не происходит обновления значений пилообразных сигналов ни по одной линии за заданное время (вход **tpila**), то фиксируется потеря связи между ПЛК (выход **conn_fault = 1**);
- Наличие связи фиксируется (выход **conn_fault = 0**) с задержкой времени (вход **trecon**) после того, когда снова начинают изменяться значения передаваемых пилообразных сигналов;
- Текущие значения пилообразного сигнала выдаются на выходы: **L_pila** и **R_pila_1**, **R_pila_2**, соответственно, собственный сигнал и сигналы от соседнего контроллера по двум линиям связи.

Для обмена двумя диагностическими сигналами в проекте контроллера должно быть добавлено два менеджера **SyncMan**, каждый из которых связывается по своему порту.

Выход **L_Master** используется в прикладной программе для задания признака ведущего у линейки ввода/вывода. А его инверсия – как признак необходимости синхронизации (входы **sync** блоков **SyncMan**). Другие выходы блока могут использоваться для диагностики и в пользовательской логике.



ПРИМЕЧАНИЕ

Блок **MasterSel** является составным, поэтому подробно логику его работы можно посмотреть на внутренних страницах. Для этого следует открыть библиотеку **paSync** в представлении **Дерево**.

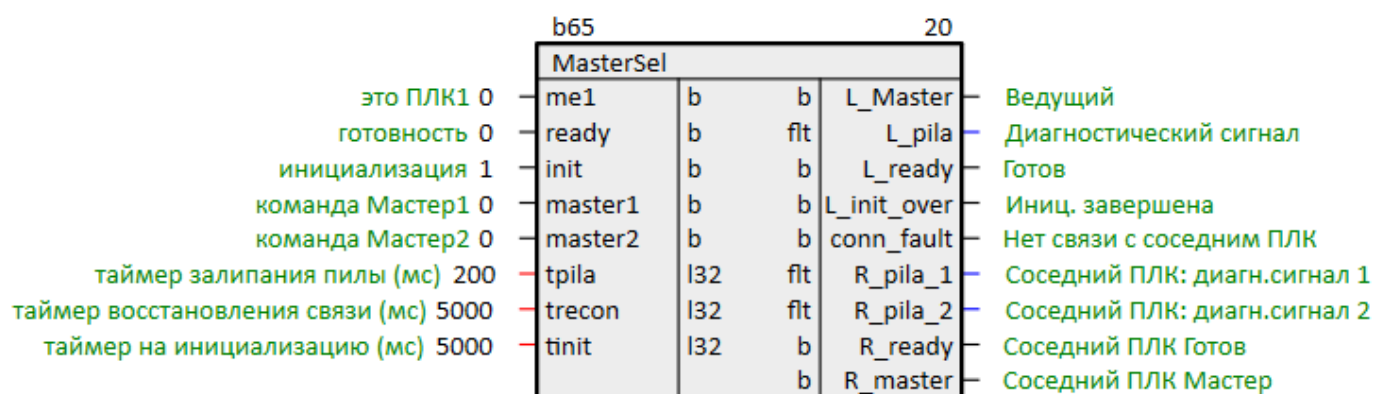


Рисунок 2.5 – Определение ведущего (MasterSel)

Пример работы с блоком приведен в [разделе 5](#).

3 Создание синхронизированных модулей

Для реализации синхронизации модулей двух контроллеров следует дублировать их места работы. Для этого модули контроллеров должны находиться в одном проекте.

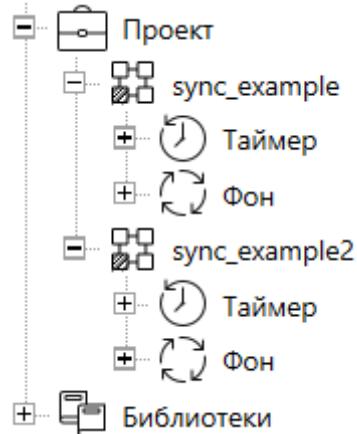


Рисунок 3.1 – Дерево проекта с двумя модулями

Для того чтобы дублировать место работы одного модуля следует:

1. Захватить место работы мышью и перетащить на второй модуль.
2. В выпадающем меню выбрать **Добавить**. Места работы (у первого и у второго модуля) подсвечиваются желтым.
3. Аналогичное место работы у второго модуля следует удалить.

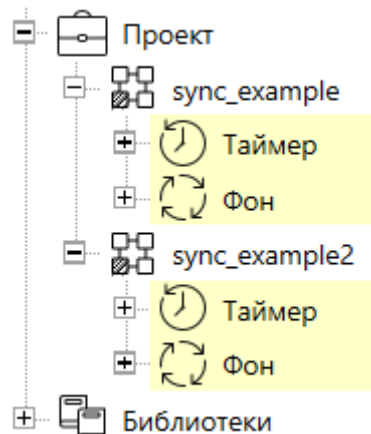


Рисунок 3.2 – Дерево проекта. Дублированные модули

Теперь все изменения в местах работы одного модуля будут дублироваться во втором модуле и наоборот.



ВНИМАНИЕ

Важно понимать, что не только сами блоки и их расположение будет дублироваться во второй модуль, но и значения инициализации на входах блоков, комментарии, связи.

Для удобства перемещения по дублированным страницам модулей можно воспользоваться командой **Показать**. Вызвать ее можно в выпадающем меню при нажатии ПКМ на свободном месте страницы.

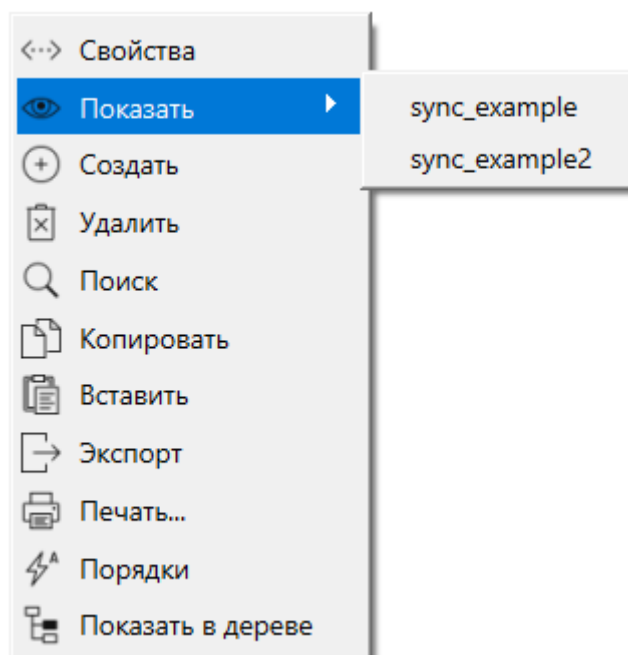


Рисунок 3.3 – Команда Показать

При работе с синхронизированными модулями для блоков, реализующих протоколы обмена, таких как **OpсUAServer** и **OpсUAClient** из библиотеки **paOpсUA**, **TcplpSrA** и **TcplpCIA** из библиотеки **paCore** и др., при задании входов следует использовать SQL-запросы к соответствующим свойствам модуля.

Это необходимо для уникальных параметров контроллеров, таких как IP-адреса. Для задания параметров «соседнего» контроллера рекомендуется использовать пользовательские свойства.

Запрос IP-адреса (**prop_ip**):

```
"<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_ip"</sql>"
```

Запрос номера порта отладчика (**prop_debug_port**):

```
<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_debug_port"</sql>
```

Запрос пользовательского свойства **Пользовательское свойство 00 (prop_0)**:

```
<sql> SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_0"</sql>
```

4 Синхронизация модулей. Менеджер синхронизации (SyncMan)

В [разделе 3](#) была рассмотрена синхронизация проектов на этапе разработки. Синхронизация проектов во время исполнения выполняется блоком менеджер синхронизации **SyncMan** для синхронизирующихся блоков из библиотеки **paSync**.

Синхронизация исполняемых проектов осуществляется по протоколу **OPC UA**.

Блок **SyncMan** является модифицированным блоком **OPC UA**-клиента (блок **OpcUAClient** из библиотеки **paOpcUA**), реализующим одну подписку к серверу.

Каждый ПЛК, программируемый в **Полигон**, является **OPC UA**-сервером, так как **Отладчик** среды подключается к контроллеру как **OPC UA**-клиент. Преднастроенный **OPC UA**-сервер (блок **OpcUAServer** из библиотеки **paOpcUA**) добавляется автоматически при создании модуля из шаблона **Модуль с отладчиком для контроллера** в месте работы **Фон**, программа **Debug**.

Подробно реализация протокола **OPC UA** в среде **Полигон** описана в документе [Обмен с верхним уровнем. Библиотека paOpcUA](#).

Далее рассмотрена синхронизация программ в контроллерах на основе проекта, созданного в [разделе 3](#). Для настройки следует:

1. Добавить на страницу таймера любого из дублированных модулей пару блоков из библиотеки **paSync**. Убедиться, что на аналогичной странице второго модуля изменения дублировались.

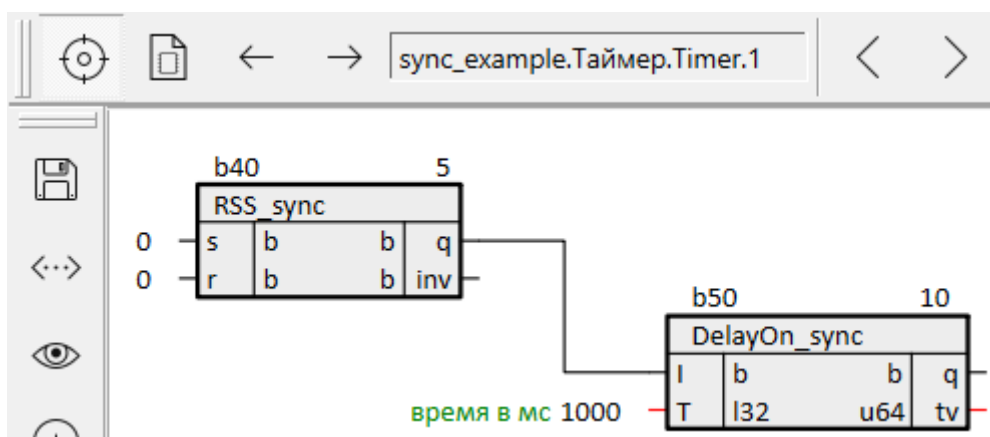


Рисунок 4.1 – Страница модуля

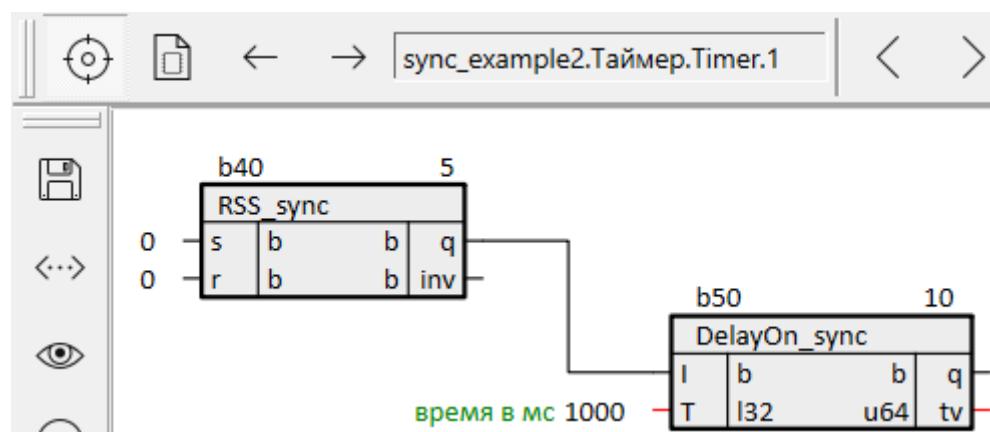


Рисунок 4.2 – Страница «соседнего» модуля

2. На любую страницу в месте работы **Фон** добавить блок **SyncMan**.
3. На входы блока **lip**, **lprt**, **rip**, **rprt** подать SQL-запросы к свойствам модуля. Примеры SQL-запросов приведены в [разделе 3](#). В данном примере вход **fnum = 0**.

В данном примере блок **SyncMan** будет подключаться к **OPC UA**-серверу отладчика, при необходимости можно создать отдельный **OPC UA**-сервер, выделенный для синхронизации проектов.


```

ip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_ip"</sql>"
prt = <sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_debug_port"</sql>
    
```

		b19		5		
OpcUAServer						
"[SQL]"	ip	str	i32	st		0 - running, 1 - failed (см. OPCUA Part5 12.6)
порт [SQL]	prt	i16	uch	cn		активных соединений
выбор сетевой платы / или /sock2 "/"	sdr	str	uch	cnDbg		соединений с отладчиком
0 - running, 1 - failed (см. OPCUA Part5 12.6)	st	i32	uch	cnPA		соединений с клиентом PA
255 - ведущий, 199 - ведомый (см. OPCUA Part5 6.6.2.4.2)	sl	uch	uch	cnOth		соединений с другими клиентами
макс. кол-во соединений	max	uch				

```

lip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_ip"</sql>"
lprt = <sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_0"</sql> (Пользовательское свойство 00)
rip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_1"</sql>" (Пользовательское свойство 01)
rprt = <sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_2"</sql> (Пользовательское свойство 02)
    
```

		b59		10		
SyncMan						
связь: 0 - выкл, 1 - вкл	1	enb	u8	i32	sts	0 - нет обмена, 1 - обмен, 2 - в резерве, >2 - нег
синхронизация: 0 - выкл, 1 - вкл	0	sync	u8	u8	svld	синхронизация работает
таймаут (мс)	500	wait	u132	i32	sst	статус сервера
локальный IP адрес "[SQL]"		lip	str	u8	ssl	service level сервера
локальный порт [SQL]		lprt	i16	u32	sid	ID подписки
выбор сетевой платы / или /sock2 "/"		sdr	str	u32	ssn	номер уведомления подписки
IP адрес сервера "[SQL]"		rip	str	u32	rcnt	принято
порт сервера [SQL]		rprt	i16	u32	wcnt	отправлено
логин "admin"		usr	str	u8	prio	приоритет дополнительного потока (0 - отключ
пароль "1"		psw	str	u32	dsz	количество данных
приоритет потока (1-200), 0 - отключить	30	prio	u8	flt	wrk	текущее время работы (мс)
сброс временных счетчиков	0	rst	u8	flt	mwrk	максимальное время работы (мс)
				flt	scan	текущее время получения данных (мс)
			#	flt	mscan	максимальное время получения данных (мс)
				u16	diag	диагностический счетчик

Рисунок 4.3 – Настройка OPC UA-сервера и SyncMan

В данном примере в качестве «соседнего» контроллера будет выступать виртуальный контроллер.

sync_example (модуль) x

Свойство	Значение
IP адрес	10.2.7.77
SSH: логин	root
SSH: пароль	owen
Имя	sync_example
Номер	0
ОС	Linux Овен прошивка 3.x
Пароль admin	<password>
Подключаться через	SSH
Пользовательское свойство 00	8000
Пользовательское свойство 01	10.2.3.179
Пользовательское свойство 02	4840
Порт отладчика	4840

Добавление новых свойств:

привязать к родителю

Рисунок 4.4 – Свойства модуля

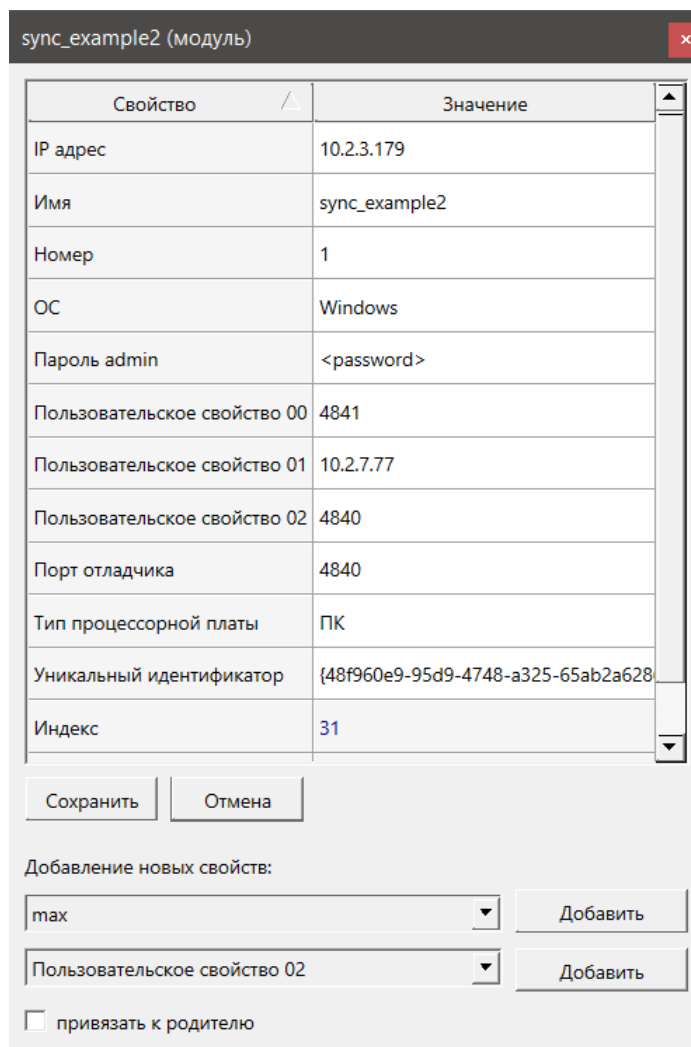


Рисунок 4.5 – Свойства «соседнего» модуля

4. Запустить программы на обоих контроллерах.

У блоков **OPC UA**-серверов в обоих модулях отобразится активное подключение клиента ПА – блока **SyncMan**.

b19		5					
OpcUAServer		80мкс					
"[SQL]"	10.2.7.77	ip	str	i32	st	0	0 - running, 1 - failed (см. OPCUA Part5 12.6)
порт [SQL]	4840	prt	i16	uch	cn	2	активных соединений
/ или /sock2	"/" /	sdr	str	uch	cnDbg	1	соединений с отладчиком
JA Part5 12.6)	0 0	st	i32	uch	cnPA	1	соединений с клиентом ПА
i.6.2.4.2)	255 255	sl	uch	uch	cnOth	0	соединений с другими клиентами
динений	255 255	max	uch				

Рисунок 4.6 – OPC UA-сервер. Подписка SyncMan «соседнего» контроллера

После запуска входы **sync** блоков **SyncMan** в обоих модулях равны **0**. Синхронизации не происходит.

5. Установить в одном из модулей **sync = 1** – данный контроллер станет ведомым. Его блоки будут синхронизироваться с соответствующими блоками ведущего контроллера.

Выход блока **svld** установится в **1** – синхронизация работает.

b59		SyncMan		10		1мкс		
связь: 0 - выкл, 1 - вкл	1 1	enb	u8	i32	sts	1	0 - нет обмена, 1 - обмен	
синхронизация: 0 - выкл, 1 - вкл	0 1	sync	u8	u8	svld	1	синхронизация работает	
таймаут (мс)	500 500	wait	ul32	i32	sst	0	статус сервера	
локальный IP адрес "[SQL]"	10.2.3.179	lip	str	u8	ssl	255	service level сервера	
локальный порт [SQL]	-715	lprt	i16	u32	sid	1	ID подписки	
номер сетевой платы / или /sock2 "/" /	/ /	sdr	str	u32	ssn	18802	номер уведомления	
IP адрес сервера "[SQL]"	10.2.7.77	rip	str	u32	rcnt	18926	принято	
порт сервера [SQL]	4840	rprt	i16	u32	wcnt	18926	отправлено	
логин "admin"	admin	usr	str	u8	prio	30	приоритет дополнительный	
пароль "1"	1	psw	str	u32	dsz	3	количество данных	
размер потока (1-200), 0 - отключить	30 30	prio	u8	flt	wrk	0	текущее время работы (мс)	
сброс временных счетчиков	0 0	rst	u8	flt	mwrk	159	максимальное время работы (мс)	
				#	scan	32	текущее время получения	
					flt	mscan	64	максимальное время получения
					u16	diag	37871	диагностический счетчик

Рисунок 4.7 – SyncMan. Синхронизация включена – этот контроллер ведомый

6. Взвести триггер на странице ведущего контроллера.

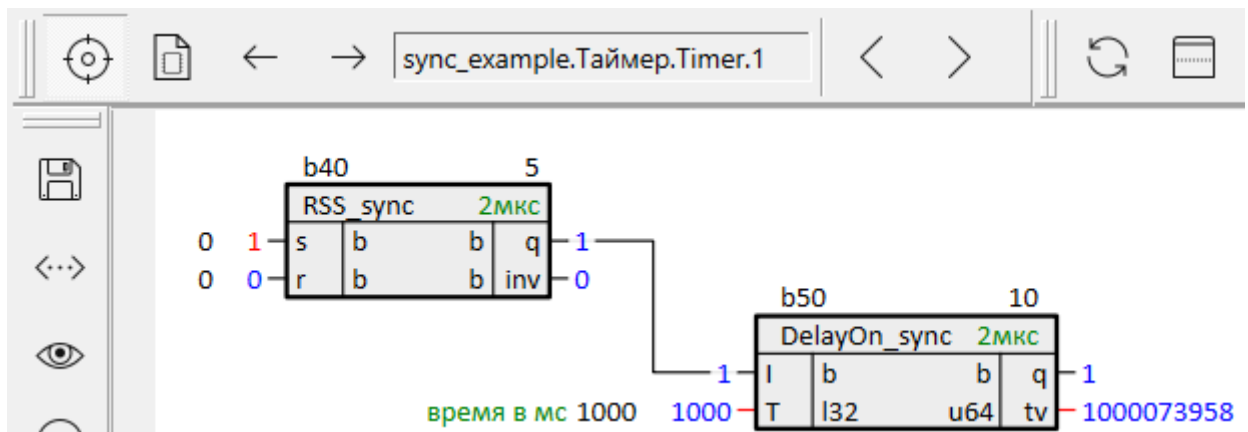


Рисунок 4.8 – Изменение выходов блоков со стороны ведущего контроллера

7. Проверить, что блок триггера на странице ведомого контроллера синхронизировался с блоком ведущего.

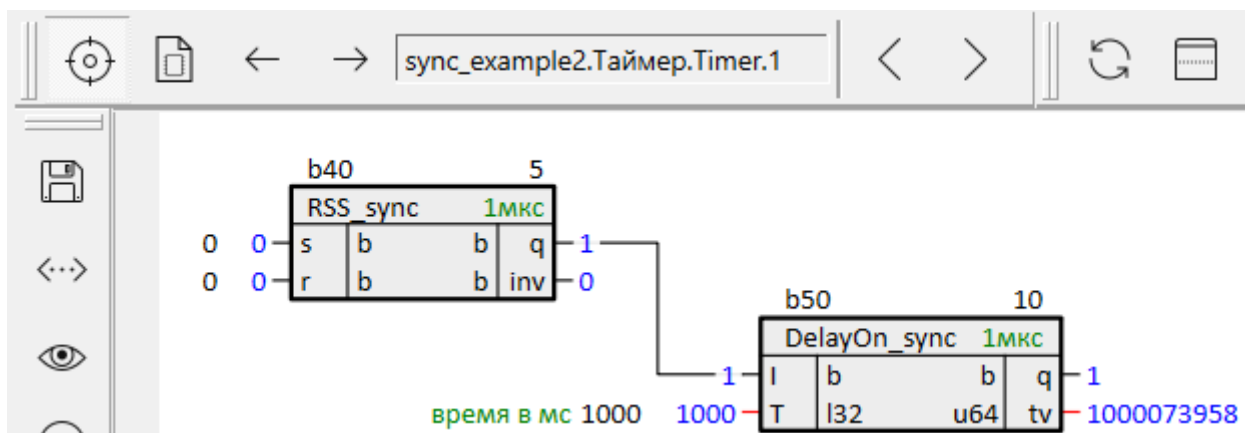


Рисунок 4.9 – Синхронизация блока ведомого контроллера

Если попытаться сбросить значение на выходе триггера у ведомого контроллера, подав 1 на вход r, – выход не сбросится.

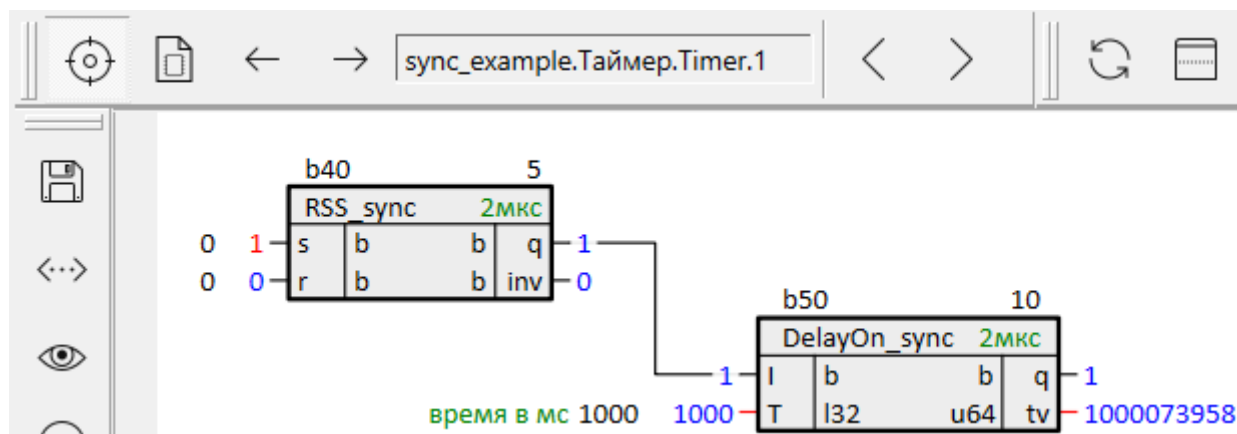


Рисунок 4.10 – Изменение со стороны ведомого контроллера не принимается

Помимо блоков с поддержкой синхронизации синхронизировать можно отдельные входы/выходы в модулях. Для этого следует добавить необходимые входы/выходы в раздел **Данные** внутри блока **SyncMan**.

См. описание синхронизации объектов данных из раздела **Данные** в [разделе 2.1](#).

Добавить объект данных в раздел можно одним из следующих способов:

1. Открыть на одной странице блок **SyncMan**, на другой странице блок с входом/выходом, который необходимо добавить. Выделить вход/выход и с нажатым **Ctrl** перетащить его на блок **SyncMan**. Отпустить, выбрать команду **Добавить**.
2. Открыть блок **SyncMan** в дереве (со страницы это проще всего сделать командой **Показать в дереве**), раскрыть его. Вход/выход перетащить в раздел **Данные**, выбрать команду **Добавить**.

5 Определение ведущего контроллера (MasterSel)

В [разделе 2.1](#) было рассмотрено ручное включение синхронизации модулей, путем установки значения входа **sync** у блоков **SyncMan**.

Для автоматического определения ролей ведущего/ведомого контроллера можно воспользоваться готовым блоком **MasterSel** или написать свой алгоритм.

MasterSel предназначен для определения роли ведущего между двумя контроллерами. Контроль наличия связи между контроллерами осуществляется посредством передачи пилообразного сигнала по двум линиям связи. Линии связи организуются с помощью двух блоков **SyncMan** в проекте, каждый из которых осуществляет связь по своему порту.

Рассмотрим примеры реализации резервированной пары контроллеров на основе блока **MasterSel**.

В проекте должно быть создано два синхронизированных модуля. Создание синхронизированных модулей было рассмотрено в [разделе 3](#).

5.1 Пример реализации резервирования ПЛК с общими Mx210

Далее рассмотрен пример реализации резервированной пары контроллеров с общей корзиной модулей серии **Mx210**. Данную схему удобно настраивать с помощью **Мастера настройки** в web-конфигураторе ПЛК (см. [схему 4](#) в [Руководстве по эксплуатации](#)).



ПРИМЕЧАНИЕ

Настройка обмена в среде Полигон для [схемы 5](#) (см. [Руководство по эксплуатации](#)) принципиально не отличается от рассмотренной в данном разделе.

Для организации двух линий связи между контроллерами использованы интерфейсы **LAN** и **REDU**, интерфейс **P3** использован для подключения отладчиком среды **Полигон**.

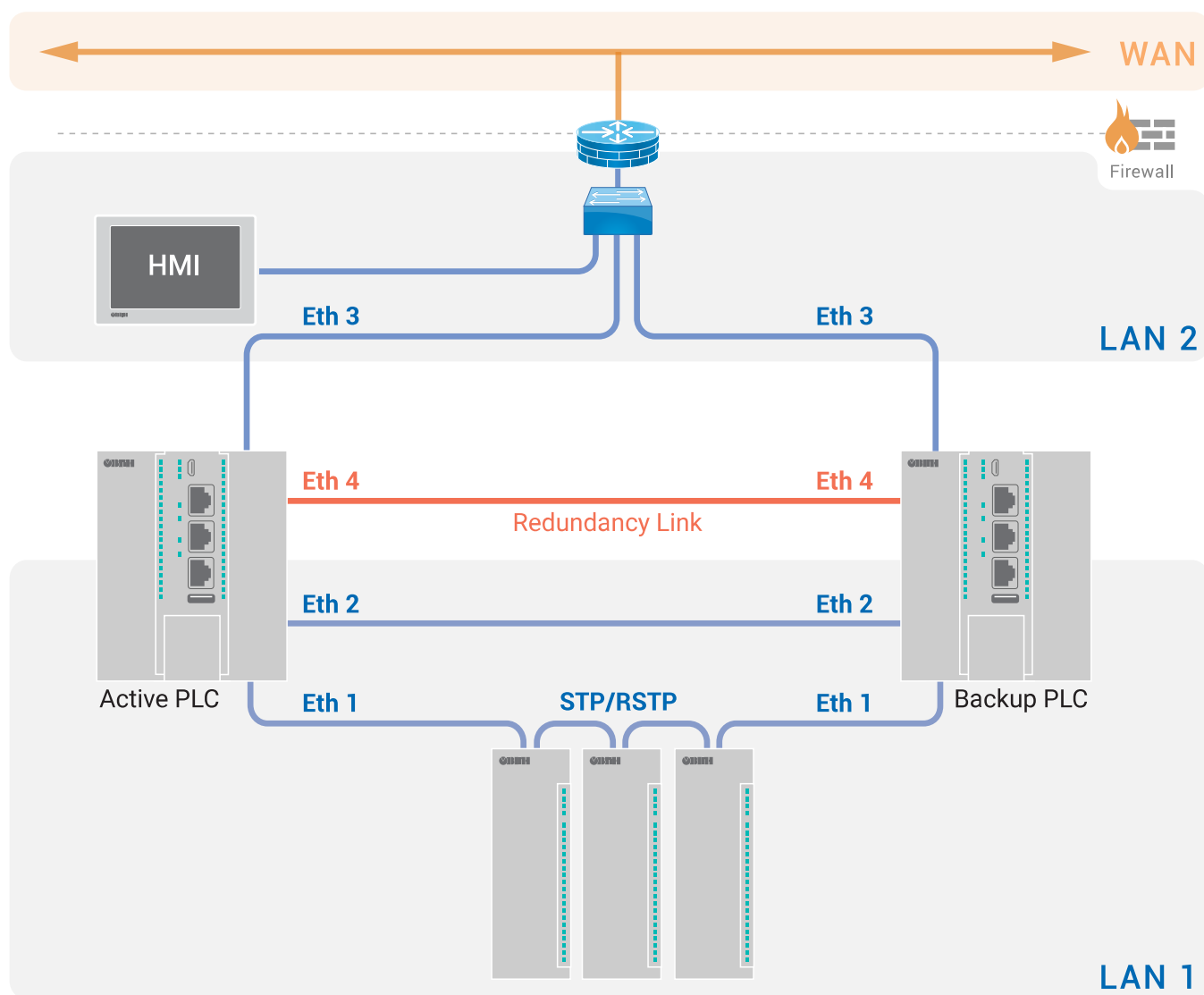


Рисунок 5.1 – Структурная схема примера

Таблица 5.1 – Настройки интерфейсов контроллеров в примере

Контроллер	Интерфейс LAN Ethernet 1...2	Интерфейс P3 Ethernet 3	Интерфейс REDU Ethernet 4
ПЛК1	192.168.0.12/16 RSTP	DHCP-клиент	192.168.10.12/24
ПЛК2	192.168.0.14/16 RSTP	DHCP-клиент	192.168.10.14/24

Для организации двух линий связи следует:

1. Добавить в проект два **OPC UA**-сервера – блоки **OpcUAServer** из библиотеки **paOpcUA**.
2. Настроить **OPC UA**-серверы в соответствии с таблицей 5.1 с помощью SQL-запросов к свойствам модуля – **Пользовательское свойство 00(LAN)** и **Пользовательское свойство 01 (REDU)**. Примеры SQL-запросов приведены в разделе 3.

ip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_0"</sql>" (Пользовательское свойство 00)

192.168.0.X/16 "[SQL]"
порт 4841

выбор сетевой платы / или /sock2 "/"
0 - running, 1 - failed (см. OPCUA Part5 12.6) 0
255 - ведущий, 199 - ведомый (см. OPCUA Part5 6.6.2.4.2) 255
макс. кол-во соединений 255

OPC UA-сервер 1 5				
OpcUAServer				
ip	str	i32	st	0 - running, 1 - failed (см. OPCUA Part5 12.6)
prt	i16	uch	cn	активных соединений
sdr	str	uch	cnDbg	соединений с отладчиком
st	i32	uch	cnPA	соединений с клиентом PA
sl	uch	uch	cnOth	соединений с другими клиентами
max	uch			

ip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_1"</sql>" (Пользовательское свойство 01)

192.168.10.X/24 "[SQL]"
порт 4842

выбор сетевой платы / или /sock2 "/"
0 - running, 1 - failed (см. OPCUA Part5 12.6) 0
255 - ведущий, 199 - ведомый (см. OPCUA Part5 6.6.2.4.2) 255
макс. кол-во соединений 255

OPC UA-сервер 2 10				
OpcUAServer				
ip	str	i32	st	0 - running, 1 - failed (см. OPCUA Part5 12.6)
prt	i16	uch	cn	активных соединений
sdr	str	uch	cnDbg	соединений с отладчиком
st	i32	uch	cnPA	соединений с клиентом PA
sl	uch	uch	cnOth	соединений с другими клиентами
max	uch			

Рисунок 5.2 – Настройка OPC UA-серверов: установка локальных IP-адресов

3. Добавить в проект два блока **SyncMan** (OPC UA-клиенты).
4. Настроить блоки **SyncMan** в соответствии с таблицей 5.1 с помощью SQL-запросов к свойствам модуля: для локальных адресов также используем **Пользовательское свойство 00** и **Пользовательское свойство 01**, для IP адресов соседнего контроллера зададим **Пользовательское свойство 02** и **Пользовательское свойство 03**. В данном примере входы **fnum = 0**.

lip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_0"</sql>" (Пользовательское свойство 00)
rip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_2"</sql>" (Пользовательское свойство 02)

связь: 0 - выкл, 1 - вкл 1
синхронизация: 0 - выкл, 1 - вкл 0
таймаут (мс) 500
локальный IP адрес "[SQL]"
локальный порт 8000
выбор сетевой платы / или /sock2 "/"
IP адрес сервера "[SQL]"
порт сервера 4841
логин "admin"
пароль "1"
приоритет потока (1-200), 0 - отключить 30
сброс временных счетчиков 0

SyncMan 1 5				
SyncMan				
enb	u8	i32	sts	0 - нет обмена, 1 - обмен, 2 - в резерве, >2 - переходное состояние,
sync	u8	u8	svld	синхронизация работает
wait	ul32	i32	sst	статус сервера
lip	str	u8	ssl	service level сервера
lprt	i16	u32	sid	ID подписки
sdr	str	u32	ssn	номер уведомления подписки
rip	str	u32	rcnt	принято
rpri	i16	u32	wcnt	отправлено
usr	str	u8	prio	приоритет дополнительного потока (0 - отключен)
psw	str	u32	dsz	количество данных
prio	u8	flt	wrk	текущее время работы (мс)
rst	u8	flt	mwrk	максимальное время работы (мс)
		flt	scan	текущее время получения данных (мс)
		flt	mscan	максимальное время получения данных (мс)
		u16	diag	диагностический счетчик

lip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_1"</sql>" (Пользовательское свойство 01)
rip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_3"</sql>" (Пользовательское свойство 03)

связь: 0 - выкл, 1 - вкл 1
синхронизация: 0 - выкл, 1 - вкл 0
таймаут (мс) 500
локальный IP адрес "[SQL]"
локальный порт 8001
выбор сетевой платы / или /sock2 "/"
IP адрес сервера "[SQL]"
порт сервера 4842
логин "admin"
пароль "1"
приоритет потока (1-200), 0 - отключить 30
сброс временных счетчиков 0

SyncMan 2 10				
SyncMan				
enb	u8	i32	sts	0 - нет обмена, 1 - обмен, 2 - в резерве, >2 - переходное состояние,
sync	u8	u8	svld	синхронизация работает
wait	ul32	i32	sst	статус сервера
lip	str	u8	ssl	service level сервера
lprt	i16	u32	sid	ID подписки
sdr	str	u32	ssn	номер уведомления подписки
rip	str	u32	rcnt	принято
rpri	i16	u32	wcnt	отправлено
usr	str	u8	prio	приоритет дополнительного потока (0 - отключен)
psw	str	u32	dsz	количество данных
prio	u8	flt	wrk	текущее время работы (мс)
rst	u8	flt	mwrk	максимальное время работы (мс)
		flt	scan	текущее время получения данных (мс)
		flt	mscan	максимальное время получения данных (мс)
		u16	diag	диагностический счетчик

Рисунок 5.3 – Настройка SyncMan: установка IP адресов

5. Согласовать номера портов, выделяемых для обмена между OPC UA-серверами и клиентами контроллеров.

The screenshot shows a configuration window titled "plc1 (модуль)". It contains a table of properties and their values. Two rows are highlighted with red boxes: "Имя" (Name) with value "plc1" and a group of four "Пользовательское свойство" (User-defined property) rows with IP addresses.

Свойство	Значение
IP адрес	10.2.12.12
SSH: логин	root
SSH: пароль	owen
Имя	plc1
Номер	0
ОС	Linux Овен прошивка 3.x
Пароль admin	<password>
Подключаться через	SSH
Пользовательское свойство 00	192.168.0.12
Пользовательское свойство 01	192.168.10.12
Пользовательское свойство 02	192.168.0.14
Пользовательское свойство 03	192.168.10.14
Порт отладчика	4840
Тип процессорной платы	Овен ПЛК210

Buttons: Сохранить, Отмена

Добавление новых свойств:

max [Dropdown] [Добавить]

Пользовательское свойство 03 [Dropdown] [Добавить]

привязать к родителю

Рисунок 5.4 – Свойства модуля ПЛК1

plc2 (модуль) x

Свойство	Значение
IP адрес	10.2.14.14
SSH: логин	root
SSH: пароль	owen
Имя	plc2
Номер	1
ОС	Linux Овен прошивка 3.x
Пароль admin	<password>
Подключаться через	SSH
Пользовательское свойство 00	192.168.0.14
Пользовательское свойство 01	192.168.10.14
Пользовательское свойство 02	192.168.0.12
Пользовательское свойство 03	192.168.10.12
Порт отладчика	4840
Тип процессорной платы	Овен ПЛК210

Сохранить Отмена

Добавление новых свойств:

max Добавить

Пользовательское свойство 03 Добавить

привязать к родителю

Рисунок 5.5 – Свойства модуля ПЛК2

- Загрузить проекты на контроллеры.

ip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_0"</sql>" (Пользовательское свойство 00)

ОПC UA-сервер 1		5		
OpcUAServer		45мс		
192.168.0.X/16 "[SQL]"	192.168.0.12	ip	str i32 st	0 0 - running, 1 - failed (см. OPCUA Part5 12.6)
порт 4841	4841	prt	i16 uch cn	1 активных соединений
выбор сетевой платы / или /sock2 "/" /		sdr	str uch cnDbg	0 соединений с отладчиком
0 - running, 1 - failed (см. OPCUA Part5 12.6)	0 0	st	i32 uch cnPA	1 соединений с клиентом PA
255 - ведущий, 199 - ведомый (см. OPCUA Part5 6.6.2.4.2)	255 255	sl	uch uch cnOth	0 соединений с другими клиентами
макс. кол-во соединений	255 255	max	uch	

ip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_1"</sql>" (Пользовательское свойство 01)

ОПC UA-сервер 2		10		
OpcUAServer		42мс		
192.168.10.X/24 "[SQL]"	192.168.10.12	ip	str i32 st	0 0 - running, 1 - failed (см. OPCUA Part5 12.6)
порт 4842	4842	prt	i16 uch cn	1 активных соединений
выбор сетевой платы / или /sock2 "/" /		sdr	str uch cnDbg	0 соединений с отладчиком
0 - running, 1 - failed (см. OPCUA Part5 12.6)	0 0	st	i32 uch cnPA	1 соединений с клиентом PA
255 - ведущий, 199 - ведомый (см. OPCUA Part5 6.6.2.4.2)	255 255	sl	uch uch cnOth	0 соединений с другими клиентами
макс. кол-во соединений	255 255	max	uch	

Рисунок 5.6 – Успешный обмен: OPC UA-серверы (ПЛК1)

lip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_0"</sql>" (Пользовательское свойство 00)

rip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_2"</sql>" (Пользовательское свойство 02)

SyncMan 1		5		
SyncMan		3мс		
связь: 0 - выкл, 1 - вкл	1 1	enb	u8 i32 sts	1 0 - нет обмена, 1 - обмен 2 - в резерве, >2 - переходное состояни
синхронизация: 0 - выкл, 1 - вкл	0 0	sync	u8 u8 svld	0 синхронизация работает
таймаут (мс)	500 500	wait	ul32 i32 sst	0 статус сервера
локальный IP адрес "[SQL]"	192.168.0.14	lip	str u8 ssl	255 service level сервера
локальный порт	8000 8000	lppt	i16 u32 sid	3 ID подписки
выбор сетевой платы / или /sock2 "/" /		sdr	str u32 ssn	18107 номер уведомления подписки
IP адрес сервера "[SQL]"	192.168.0.12	rip	str u32 rcnt	18190 принято
порт сервера	4841 4841	rppt	i16 u32 wcnt	18191 отправлено
логин "admin" admin		usr	str u8 prio	30 приоритет дополнительного потока (0 - отключен)
пароль "1" 1		psw	str u32 dsz	2 количество данных
приоритет потока (1-200), 0 - отключить	30 30	prio	u8 flt wrk	0.00175 текущее время работы (мс)
сброс временных счетчиков	0 0	rst	u8 flt mwrk	120.687 максимальное время работы (мс)
			flt scan	20.3128 текущее время получения данных (мс)
			flt mscan	41.1924 максимальное время получения данных (мс)
			u16 diag	5354 диагностический счетчик

lip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_1"</sql>" (Пользовательское свойство 01)

rip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_3"</sql>" (Пользовательское свойство 03)

SyncMan 2		10		
SyncMan		1мс		
связь: 0 - выкл, 1 - вкл	1 1	enb	u8 i32 sts	1 0 - нет обмена, 1 - обмен 2 - в резерве, >2 - переходное состояни
синхронизация: 0 - выкл, 1 - вкл	0 0	sync	u8 u8 svld	0 синхронизация работает
таймаут (мс)	500 500	wait	ul32 i32 sst	0 статус сервера
локальный IP адрес "[SQL]"	192.168.10.14	lip	str u8 ssl	255 service level сервера
локальный порт	8001 8001	lppt	i16 u32 sid	1 ID подписки
выбор сетевой платы / или /sock2 "/" /		sdr	str u32 ssn	26817 номер уведомления подписки
IP адрес сервера "[SQL]"	192.168.10.12	rip	str u32 rcnt	26936 принято
порт сервера	4842 4842	rppt	i16 u32 wcnt	26949 отправлено
логин "admin" admin		usr	str u8 prio	30 приоритет дополнительного потока (0 - отключен)
пароль "1" 1		psw	str u32 dsz	2 количество данных
приоритет потока (1-200), 0 - отключить	30 30	prio	u8 flt wrk	0.000584 текущее время работы (мс)
сброс временных счетчиков	0 0	rst	u8 flt mwrk	120.862 максимальное время работы (мс)
			flt scan	20.3181 текущее время получения данных (мс)
			flt mscan	46.8431 максимальное время получения данных (мс)
			u16 diag	5326 диагностический счетчик

Рисунок 5.7 – Успешный обмен: OPC UA-клиенты SyncMan (ПЛК2)

ip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_0"</sql>" (Пользовательское свойство 00)

ОПC UA-сервер 1 5

OpcUAServer		44мс			
ip	str	i32	st	0	0 - running, 1 - failed (см. OPCUA Part5 12.6)
prt	i16	uch	cn	1	активных соединений
sdr	str	uch	cnDbg	0	соединений с отладчиком
st	i32	uch	cnPA	1	соединений с клиентом PA
sl	uch	uch	cnOth	0	соединений с другими клиентами
max	uch				

192.168.0.X/16 "[SQL]" 192.168.0.14
порт 4841 4841
выбор сетевой платы / или /sock2 "/" /
0 - running, 1 - failed (см. OPCUA Part5 12.6) 0 0
255 - ведущий, 199 - ведомый (см. OPCUA Part5 6.6.2.4.2) 255 255
макс. кол-во соединений 255 255

ip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_1"</sql>" (Пользовательское свойство 01)

ОПC UA-сервер 2 10

OpcUAServer		41мс			
ip	str	i32	st	0	0 - running, 1 - failed (см. OPCUA Part5 12.6)
prt	i16	uch	cn	1	активных соединений
sdr	str	uch	cnDbg	0	соединений с отладчиком
st	i32	uch	cnPA	1	соединений с клиентом PA
sl	uch	uch	cnOth	0	соединений с другими клиентами
max	uch				

192.168.10.X/24 "[SQL]" 192.168.10.14
порт 4842 4842
выбор сетевой платы / или /sock2 "/" /
0 - running, 1 - failed (см. OPCUA Part5 12.6) 0 0
255 - ведущий, 199 - ведомый (см. OPCUA Part5 6.6.2.4.2) 255 255
макс. кол-во соединений 255 255

Рисунок 5.8 – Успешный обмен: ОПC UA-серверы (ПЛК2)

lip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_0"</sql>" (Пользовательское свойство 00)

rip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_2"</sql>" (Пользовательское свойство 02)

SyncMan 1 5

SyncMan		5мс			
enb	u8	i32	sts	1	0 - нет обмена, 1 - обмен 2 - в резерве, >2 - переходное состояни
sync	u8	u8	svld	0	синхронизация работает
wait	u132	i32	sst	0	статус сервера
lip	str	u8	ssl	255	service level сервера
lprt	i16	u32	sid	3	ID подписки
sdr	str	u32	ssn	6629	номер уведомления подписки
rip	str	u32	rcnt	6661	принято
rpri	i16	u32	wcnt	6677	отправлено
usr	str	u8	prio	30	приоритет дополнительного потока (0 - отключен)
psw	str	u32	dsz	2	количество данных
prio	u8	flt	wrk	0.001458	текущее время работы (мс)
rst	u8	flt	mwrk	122.142	максимальное время работы (мс)
		flt	scan	2.92542	текущее время получения данных (мс)
		flt	mscan	41.298	максимальное время получения данных (мс)
		u16	diag	54496	диагностический счетчик

связь: 0 - выкл, 1 - вкл 1 1
синхронизация: 0 - выкл, 1 - вкл 0 0
таймаут (мс) 500 500
локальный IP адрес "[SQL]" 192.168.0.12
локальный порт 8000 8000
выбор сетевой платы / или /sock2 "/" /
IP адрес сервера "[SQL]" 192.168.0.14
порт сервера 4841 4841
логин "admin" admin
пароль "1" 1
оритет потока (1-200), 0 - отключить 30 30
сброс временных счетчиков 0 0

lip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_1"</sql>" (Пользовательское свойство 01)

rip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_3"</sql>" (Пользовательское свойство 03)

SyncMan 2 10

SyncMan		1мс			
enb	u8	i32	sts	1	0 - нет обмена, 1 - обмен 2 - в резерве, >2 - переходное состояни
sync	u8	u8	svld	0	синхронизация работает
wait	u132	i32	sst	0	статус сервера
lip	str	u8	ssl	255	service level сервера
lprt	i16	u32	sid	1	ID подписки
sdr	str	u32	ssn	21560	номер уведомления подписки
rip	str	u32	rcnt	21656	принято
rpri	i16	u32	wcnt	21657	отправлено
usr	str	u8	prio	30	приоритет дополнительного потока (0 - отключен)
psw	str	u32	dsz	2	количество данных
prio	u8	flt	wrk	0.000875	текущее время работы (мс)
rst	u8	flt	mwrk	121.549	максимальное время работы (мс)
		flt	scan	20.3452	текущее время получения данных (мс)
		flt	mscan	40.8473	максимальное время получения данных (мс)
		u16	diag	54582	диагностический счетчик

связь: 0 - выкл, 1 - вкл 1 1
синхронизация: 0 - выкл, 1 - вкл 0 0
таймаут (мс) 500 500
локальный IP адрес "[SQL]" 192.168.10.12
локальный порт 8001 8001
выбор сетевой платы / или /sock2 "/" /
IP адрес сервера "[SQL]" 192.168.10.14
порт сервера 4842 4842
логин "admin" admin
пароль "1" 1
оритет потока (1-200), 0 - отключить 30 30
сброс временных счетчиков 0 0

Рисунок 5.9 – Успешный обмен: ОПC UA-клиенты SyncMan (ПЛК1)

Убедившись, что настройка обмена произведена успешно, можно приступить к настройке блока выбора текущих ролей контроллеров MasterSel. Для этого следует:

1. Добавить блок **MasterSel** его в проект.

Выход **L_Master** определяет роль контроллера: **L_Master = 0** – контроллер ведомый, **L_Master = 1** – контроллер ведущий.

- Так как сигнал с выхода **L_Master** может быть использован много раз в проекте, для удобства назначить его глобальной константой. Для этого в свойствах выхода **L_Master** следует добавить свойства **Полный алиас** и **Глобальная константа**.
- Вход блока **ready** (готовность) можно сразу задать **1** или завести другие сигналы из программы.

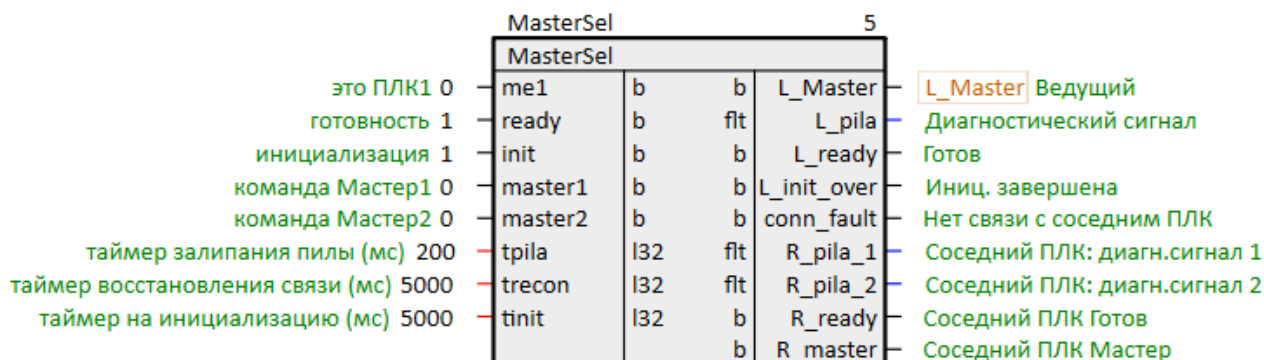


Рисунок 5.10 – Настройка MasterSel

- Провести скрытую связь между выходом **L_Master** и входами **sync** блоков **SyncMan** через блок логического НЕ – **NOT** из библиотеки **paCore**, раздел **Логические**, так как синхронизацию следует включать на ведомом контроллере, когда **L_Master = 0**.

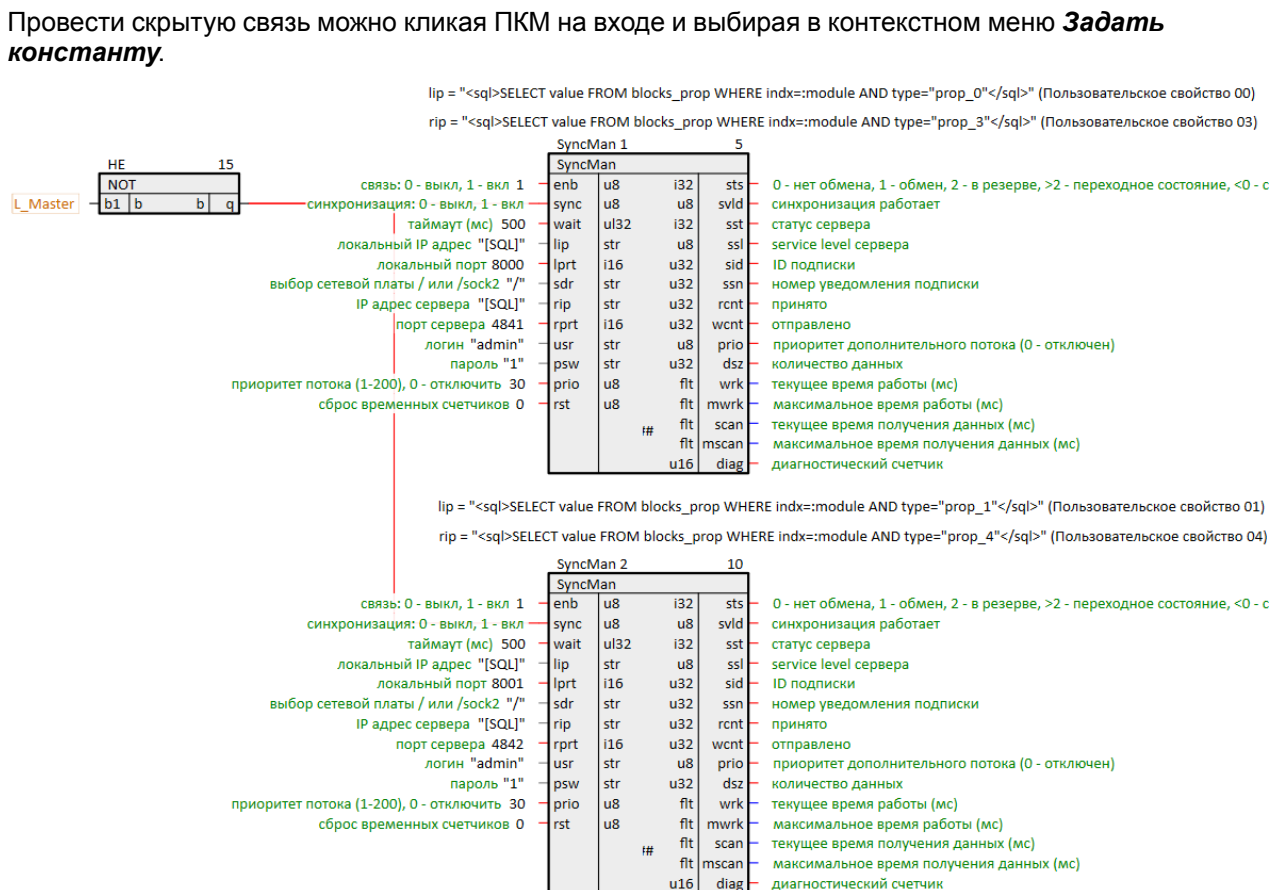


Рисунок 5.11 – Соединение блоков MasterSel с SyncMan

- Запустить программу на первом контроллере. Он станет **ведущим** по истечении времени инициализации **tinit**. На выход **L_pila** выводится диагностический сигнал контроллера. Выход **conn_fault = 1**, так как программа на втором контроллере еще не включена.

MasterSel		5	
MasterSel		79мкс	
это ПЛК1	0 0	me1	b b L_Master 1 L_Master Ведущий
готовность	1 1	ready	b flt L_pila 280 Диагностический сигнал
инициализация	1 1	init	b b L_ready 1 Готов
команда Мастер1	0 0	master1	b b L_init_over 1 Иниц. завершена
команда Мастер2	0 0	master2	b b conn_fault 1 Нет связи с соседним ПЛК
таймер залипания пилы (мс)	200 200	tpila	l32 flt R_pila_1 0 Соседний ПЛК: диагн.сигнал 1
таймер восстановления связи (мс)	5000 5000	trecon	l32 flt R_pila_2 0 Соседний ПЛК: диагн.сигнал 2
таймер на инициализацию (мс)	5000 5000	tinit	l32 b R_ready 0 Соседний ПЛК Готов
			b R_master 0 Соседний ПЛК Мастер

Рисунок 5.12 – Работа MasterSel (ПЛК1)

6. Установить $me1 = 1$ – данный контроллер имеет признак ПЛК1, он будет становиться ведущим при неопределенных условиях. На данный вход можно завести сигнал с внешней кнопки или с панели оператора и т.п.

**ПРИМЕЧАНИЕ**

На входе $me1$ можно задать константу при помощи SQL-запроса к пользовательскому свойству модуля и таким образом определить какой ПЛК является главным. Примеры запросов к свойствам модуля см. в разделе 3.

MasterSel		5	
MasterSel		71мкс	
это ПЛК1	0 1	me1	b b L_Master 1 L_Master Ведущий
готовность	1 1	ready	b flt L_pila 1920 Диагностический сигнал
инициализация	1 1	init	b b L_ready 1 Готов
команда Мастер1	0 0	master1	b b L_init_over 1 Иниц. завершена
команда Мастер2	0 0	master2	b b conn_fault 1 Нет связи с соседним ПЛК
таймер залипания пилы (мс)	200 200	tpila	l32 flt R_pila_1 0 Соседний ПЛК: диагн.сигнал 1
таймер восстановления связи (мс)	5000 5000	trecon	l32 flt R_pila_2 0 Соседний ПЛК: диагн.сигнал 2
таймер на инициализацию (мс)	5000 5000	tinit	l32 b R_ready 0 Соседний ПЛК Готов
			b R_master 0 Соседний ПЛК Мастер

Рисунок 5.13 – Работа MasterSel (ПЛК1): задание признака ПЛК1

7. Запустить программу на втором контроллере. Он станет **ведомым** по истечении времени инициализации $tinit$, так как он имеет признак ПЛК2, а соседний контроллер имеет признак ПЛК1.

MasterSel		5	
MasterSel		77мкс	
это ПЛК1	0 0	me1	b b L_Master 0 L_Master Ведущий
готовность	1 1	ready	b flt L_pila 2960 Диагностический сигнал
инициализация	1 1	init	b b L_ready 1 Готов
команда Мастер1	0 0	master1	b b L_init_over 1 Иниц. завершена
команда Мастер2	0 0	master2	b b conn_fault 0 Нет связи с соседним ПЛК
таймер залипания пилы (мс)	200 200	tpila	l32 flt R_pila_1 800.001 Соседний ПЛК: диагн.сигнал 1
таймер восстановления связи (мс)	5000 5000	trecon	l32 flt R_pila_2 840.001 Соседний ПЛК: диагн.сигнал 2
таймер на инициализацию (мс)	5000 5000	tinit	l32 b R_ready 1 Соседний ПЛК Готов
			b R_master 1 Соседний ПЛК Мастер

Рисунок 5.14 – Работа MasterSel (ПЛК2)

На выходах R_pila_1 и R_pila_2 блока *MasterSel* у обоих контроллеров отобразится диагностический сигнал от соседнего контроллера. На выходе R_ready появится 1 – оба контроллера видят, что сосед готов.

На выходе R_Master ведомого контроллера появится 1 – ПЛК2 видит, что сосед – ведущий контроллер.

MasterSel		5		
это ПЛК1	0 1	me1	b b L_Master	1 L_Master Ведущий
готовность	1 1	ready	b flt L_pila	8160 Диагностический сигнал
инициализация	1 1	init	b b L_ready	1 Готов
команда Мастер1	0 0	master1	b b L_init_over	1 Иниц. завершена
команда Мастер2	0 0	master2	b b conn_fault	0 Нет связи с соседним ПЛК
таймер залипания пилы (мс)	200 200	tpila	l32 flt R_pila_1	5880 Соседний ПЛК: диагн.сигнал 1
таймер восстановления связи (мс)	5000 5000	trecon	l32 flt R_pila_2	5840 Соседний ПЛК: диагн.сигнал 2
таймер на инициализацию (мс)	5000 5000	tinit	l32 b R_ready	1 Соседний ПЛК Готов
			b R_master	0 Соседний ПЛК Мастер

Рисунок 5.15 – Работа MasterSel (ПЛК1)

MasterSel		5		
это ПЛК1	0 0	me1	b b L_Master	0 L_Master Ведущий
готовность	1 1	ready	b flt L_pila	5880 Диагностический сигнал
инициализация	1 1	init	b b L_ready	1 Готов
команда Мастер1	0 0	master1	b b L_init_over	1 Иниц. завершена
команда Мастер2	0 0	master2	b b conn_fault	0 Нет связи с соседним ПЛК
таймер залипания пилы (мс)	200 200	tpila	l32 flt R_pila_1	3080 Соседний ПЛК: диагн.сигнал 1
таймер восстановления связи (мс)	5000 5000	trecon	l32 flt R_pila_2	3120 Соседний ПЛК: диагн.сигнал 2
таймер на инициализацию (мс)	5000 5000	tinit	l32 b R_ready	1 Соседний ПЛК Готов
			b R_master	1 Соседний ПЛК Мастер

Рисунок 5.16 – Работа MasterSel (ПЛК2)

Входы **master1** и **master2** блока **MasterSel** отвечают за ручную смену ролей контроллеров. На них можно завести сигналы с внешних кнопок или с панели оператора и т.п.

Для того чтобы передать роль ведущего от ПЛК1 к ПЛК2 следует подать 1 на вход **master2**.

MasterSel		5		
это ПЛК1	0 1	me1	b b L_Master	0 L_Master Ведущий
готовность	1 1	ready	b flt L_pila	2080 Диагностический сигнал
инициализация	1 1	init	b b L_ready	1 Готов
команда Мастер1	0 0	master1	b b L_init_over	1 Иниц. завершена
команда Мастер2	0 1	master2	b b conn_fault	0 Нет связи с соседним ПЛК
таймер залипания пилы (мс)	200 200	tpila	l32 flt R_pila_1	5520 Соседний ПЛК: диагн.сигнал 1
таймер восстановления связи (мс)	5000 5000	trecon	l32 flt R_pila_2	5520 Соседний ПЛК: диагн.сигнал 2
таймер на инициализацию (мс)	5000 5000	tinit	l32 b R_ready	1 Соседний ПЛК Готов
			b R_master	1 Соседний ПЛК Мастер

Рисунок 5.17 – Работа MasterSel (ПЛК1): передача роли ведущего контроллера

ПЛК2 станет ведущим:

MasterSel		5		
это ПЛК1	0 0	me1	b b L_Master	1 L_Master Ведущий
готовность	1 1	ready	b flt L_pila	4240 Диагностический сигнал
инициализация	1 1	init	b b L_ready	1 Готов
команда Мастер1	0 0	master1	b b L_init_over	1 Иниц. завершена
команда Мастер2	0 0	master2	b b conn_fault	0 Нет связи с соседним ПЛК
таймер залипания пилы (мс)	200 200	tpila	l32 flt R_pila_1	520.001 Соседний ПЛК: диагн.сигнал 1
таймер восстановления связи (мс)	5000 5000	trecon	l32 flt R_pila_2	560.001 Соседний ПЛК: диагн.сигнал 2
таймер на инициализацию (мс)	5000 5000	tinit	l32 b R_ready	1 Соседний ПЛК Готов
			b R_master	0 Соседний ПЛК Мастер

Рисунок 5.18 – Работа MasterSel (ПЛК2): получение роли ведущего контроллера

Условия автоматической смены ролей прописаны в разделе 2.2 и в справке среды на блок **MasterSel**.

**ПРИМЕЧАНИЕ**

Блок **MasterSel** является составным, поэтому подробно логику его работы можно посмотреть на внутренних страницах. Для этого следует открыть библиотеку **paSync** в представлении **Дерево**.

Выход **L_Master** блока **MasterSel** можно вывести на внешние сигнальные лампы, панель оператора, а также разрешающие входы протоколов обмена данными.

Рассмотрим организацию обмена с модулем линейки **Mx210**. В данном примере используется модуль **MK210-311** (схема подключения представлена на [рисунке 5.1](#)).

Настроить обмен с модулем по **Modbus TCP**.

Подробно настройка обмена по протоколу **Modbus** в среде Полигон рассмотрена в документе [Обмен по протоколу Modbus. Библиотека raModbus](#).

Для настройки следует завести на вход **enbl** сигнал **L_Master**.

```
lip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_0"</sql>" (Пользовательское свойство 00)
```

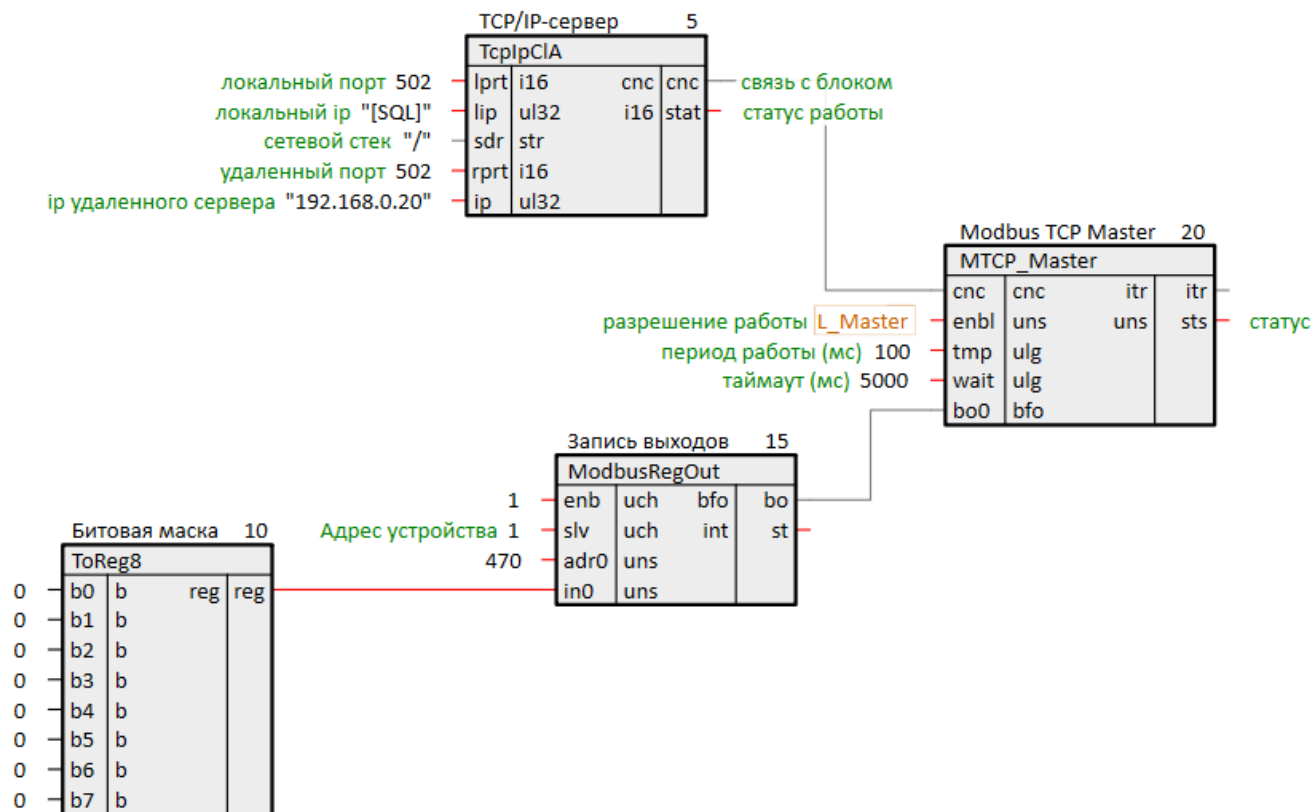


Рисунок 5.19 – Настройка обмена с модулем MK210-311

При данной настройке модуль **MK210-311** будет опрашивать тот контроллер, который в данный момент является ведущим.



ПРИМЕЧАНИЕ

Пользователь может также настроить обмен с двумя одинаковыми наборами модулей в кольце, используя программный и/или внешний аппаратный решатель (арбитр) для определения достоверного сигнала – реализация резервирования корзины модулей **Mx210**.

5.2 Пример реализации резервирования ПЛК с индивидуальными Mx210

Далее рассмотрен пример реализации резервированной пары контроллеров с индивидуальными корзинами модулей серии **Mx210**.



ПРИМЕЧАНИЕ

Настройка режимов работы сетевых интерфейсов контроллера производится в web-интерфейсе конфигурации (см. [Руководство по эксплуатации](#)).

В данном примере для организации двух линий связи между контроллерами использованы интерфейсы **P2** и **REDU**, интерфейс **P1** будет выделен для обмена с модулями **Mx210**, интерфейс **P3** будем использовать для подключения отладчиком среды **Полигон**.

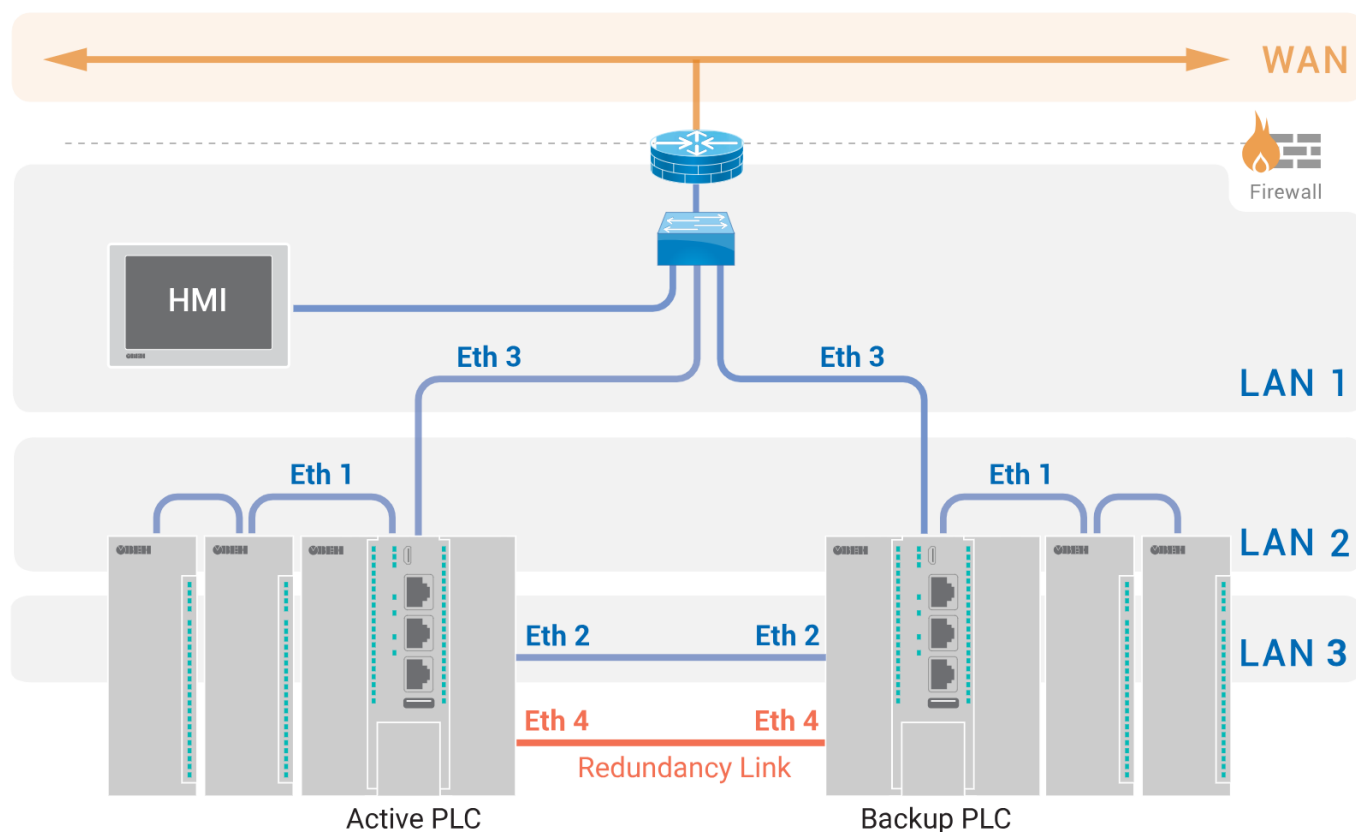


Рисунок 5.20 – Структурная схема примера

Таблица 5.2 – Настройки интерфейсов контроллеров

Контроллер	Интерфейс P1 Ethernet 1	Интерфейс P2 Ethernet 2	Интерфейс P3 Ethernet 3	Интерфейс REDU Ethernet 4
ПЛК1	192.168.0.12/16	192.168.1.12/24	DHCP-клиент	192.168.10.12/24
ПЛК2	192.168.0.14/16	192.168.1.14/24	DHCP-клиент	192.168.10.14/24

Для организации двух линий связи следует:

1. Добавить в проект два **OPC UA**-сервера – блоки **OpcUAServer** из библиотеки **paOpcUA**.
2. Настроить **OPC UA**-серверы в соответствии с таблицей 5.2 с помощью SQL-запросов к свойствам модуля – **Пользовательское свойство 00 (P2)** и **Пользовательское свойство 01 (REDU)**. Примеры SQL-запросов приведены в разделе 3.

ip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_0"</sql>" (Пользовательское свойство 00)

192.168.1.X/24 "[SQL]"

порт 4842

выбор сетевой платы / или /sock2 "/"

0 - running, 1 - failed (см. OPCUA Part5 12.6) 0

255 - ведущий, 199 - ведомый (см. OPCUA Part5 6.6.2.4.2) 255

макс. кол-во соединений 255

OPC UA-сервер 1 5

OpcUAServer			
ip	str	i32	st
prt	i16	uch	cn
sdr	str	uch	cnDbg
st	i32	uch	cnPA
sl	uch	uch	cnOth
max	uch		

0 - running, 1 - failed (см. OPCUA Part5 12.6)
активных соединений
соединений с отладчиком
соединений с клиентом PA
соединений с другими клиентами

ip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_1"</sql>" (Пользовательское свойство 01)

192.168.10.X/24 "[SQL]"

порт 4841

выбор сетевой платы / или /sock2 "/"

0 - running, 1 - failed (см. OPCUA Part5 12.6) 0

255 - ведущий, 199 - ведомый (см. OPCUA Part5 6.6.2.4.2) 255

макс. кол-во соединений 255

OPC UA-сервер 2 10

OpcUAServer			
ip	str	i32	st
prt	i16	uch	cn
sdr	str	uch	cnDbg
st	i32	uch	cnPA
sl	uch	uch	cnOth
max	uch		

0 - running, 1 - failed (см. OPCUA Part5 12.6)
активных соединений
соединений с отладчиком
соединений с клиентом PA
соединений с другими клиентами

Рисунок 5.21 – Настройка OPC UA-серверов: установка локальных IP адресов

3. Добавить в проект два блока **SyncMan** (OPC UA-клиенты).

4. Настроить блоки **SyncMan** в соответствии с **таблицей 5.2** с помощью SQL-запросов к свойствам модуля: для локальных адресов также использовать **Пользовательское свойство 00** и **Пользовательское свойство 01**, для IP адресов соседнего контроллера зададим **Пользовательское свойство 03** и **Пользовательское свойство 04**. В данном примере входы **fnum = 0**.

lip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_0"</sql>" (Пользовательское свойство 00)
 rip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_3"</sql>" (Пользовательское свойство 03)

SyncMan 1 5

SyncMan			
enb	u8	i32	sts
sync	u8	u8	svld
wait	ul32	i32	sst
lip	str	u8	ssl
lprt	i16	u32	sid
sdr	str	u32	ssn
rip	str	u32	rcnt
rprt	i16	u32	wcnt
usr	str	u8	prio
psw	str	u32	dsz
prio	u8	flt	wrk
rst	u8	flt	mwrk
		flt	scan
		flt	mscan
		u16	diag

связь: 0 - выкл, 1 - вкл 1
 синхронизация: 0 - выкл, 1 - вкл 0
 таймаут (мс) 500
 локальный IP адрес "[SQL]"
 локальный порт 8000
 выбор сетевой платы / или /sock2 "/"
 IP адрес сервера "[SQL]"
 порт сервера 4841
 логин "admin"
 пароль "1"
 приоритет потока (1-200), 0 - отключить 30
 сброс временных счетчиков 0

0 - нет обмена, 1 - обмен, 2 - в резерве, >2 - переходное состояние, <0 - с

синхронизация работает
 статус сервера
 service level сервера
 ID подписки
 номер уведомления подписки
 принято
 отправлено
 приоритет дополнительного потока (0 - отключен)
 количество данных
 текущее время работы (мс)
 максимальное время работы (мс)
 текущее время получения данных (мс)
 максимальное время получения данных (мс)
 диагностический счетчик

lip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_1"</sql>" (Пользовательское свойство 01)
 rip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_4"</sql>" (Пользовательское свойство 04)

SyncMan 2 10

SyncMan			
enb	u8	i32	sts
sync	u8	u8	svld
wait	ul32	i32	sst
lip	str	u8	ssl
lprt	i16	u32	sid
sdr	str	u32	ssn
rip	str	u32	rcnt
rprt	i16	u32	wcnt
usr	str	u8	prio
psw	str	u32	dsz
prio	u8	flt	wrk
rst	u8	flt	mwrk
		flt	scan
		flt	mscan
		u16	diag

связь: 0 - выкл, 1 - вкл 1
 синхронизация: 0 - выкл, 1 - вкл 0
 таймаут (мс) 500
 локальный IP адрес "[SQL]"
 локальный порт 8001
 выбор сетевой платы / или /sock2 "/"
 IP адрес сервера "[SQL]"
 порт сервера 4842
 логин "admin"
 пароль "1"
 приоритет потока (1-200), 0 - отключить 30
 сброс временных счетчиков 0

0 - нет обмена, 1 - обмен, 2 - в резерве, >2 - переходное состояние, <0 - с

синхронизация работает
 статус сервера
 service level сервера
 ID подписки
 номер уведомления подписки
 принято
 отправлено
 приоритет дополнительного потока (0 - отключен)
 количество данных
 текущее время работы (мс)
 максимальное время работы (мс)
 текущее время получения данных (мс)
 максимальное время получения данных (мс)
 диагностический счетчик

Рисунок 5.22 – Настройка SyncMan: установка IP адресов

5. Согласовать номера портов, выделяемых для обмена между **OPC UA**-серверами и клиентами контроллеров.

plc1 (модуль) x

Свойство	Значение
IP адрес	10.2.12.12
SSH: логин	root
SSH: пароль	owen
Имя	plc1
Номер	0
ОС	Linux Овен прошивка 3.x
Пароль admin	<password>
Подключаться через	SSH
Пользовательское свойство 00	192.168.1.12
Пользовательское свойство 01	192.168.10.12
Пользовательское свойство 02	192.168.0.12
Пользовательское свойство 03	192.168.1.14
Пользовательское свойство 04	192.168.10.14
Порт отладчика	4840
Тип процессорной платы	Овен ПЛК210

Добавление новых свойств:

привязать к родителю

Рисунок 5.23 – Свойства модуля ПЛК1

plc2 (модуль) x

Свойство	Значение
IP адрес	10.2.14.14
SSH: логин	root
SSH: пароль	owen
Имя	plc2
Номер	1
ОС	Linux Овен прошивка 3.x
Пароль admin	<password>
Подключаться через	SSH
Пользовательское свойство 00	192.168.1.14
Пользовательское свойство 01	192.168.10.14
Пользовательское свойство 02	192.168.0.14
Пользовательское свойство 03	192.168.1.12
Пользовательское свойство 04	192.168.10.12
Порт отладчика	4840
Тип процессорной платы	Овен ПЛК210

Сохранить Отмена

Добавление новых свойств:

max Добавить

Пользовательское свойство 05 Добавить

привязать к родителю

Рисунок 5.24 – Свойства модуля ПЛК2

6. Загрузить проекты в контроллеры.

```
ip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type='prop_0'</sql>" (Пользовательское свойство 00)
```

ОПC UA-сервер 1		5
OpcUAServer		44мкс
192.168.1.X/24 "[SQL]"	192.168.1.12	ip str i32 st 0 0 - running, 1 - failed (см. OPCUA Part5 12.6)
порт 4842	4842	prt i16 uch cn 1 1 активных соединений
выбор сетевой платы / или /sock2 "/" /		sdr str uch cnDbg 0 0 соединений с отладчиком
0 - running, 1 - failed (см. OPCUA Part5 12.6)	0 0	st i32 uch cnPA 1 1 соединений с клиентом PA
255 - ведущий, 199 - ведомый (см. OPCUA Part5 6.6.2.4.2)	255 255	sl uch uch cnOth 0 0 соединений с другими клиентами
макс. кол-во соединений	255 255	max uch

```
ip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type='prop_1'</sql>" (Пользовательское свойство 01)
```

ОПC UA-сервер 2		10
OpcUAServer		41мкс
192.168.10.X/24 "[SQL]"	192.168.10.12	ip str i32 st 0 0 - running, 1 - failed (см. OPCUA Part5 12.6)
порт 4841	4841	prt i16 uch cn 1 1 активных соединений
выбор сетевой платы / или /sock2 "/" /		sdr str uch cnDbg 0 0 соединений с отладчиком
0 - running, 1 - failed (см. OPCUA Part5 12.6)	0 0	st i32 uch cnPA 1 1 соединений с клиентом PA
255 - ведущий, 199 - ведомый (см. OPCUA Part5 6.6.2.4.2)	255 255	sl uch uch cnOth 0 0 соединений с другими клиентами
макс. кол-во соединений	255 255	max uch

Рисунок 5.25 – Успешный обмен: OPC UA-серверы (ПЛК1)

lip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_0"</sql>" (Пользовательское свойство 00)

rip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_3"</sql>" (Пользовательское свойство 03)

SyncMan 1		5	
SyncMan			Змкс
связь: 0 - выкл, 1 - вкл	1	enb	u8 i32 sts 1 0 - нет обмена, 1 - обмен
синхронизация: 0 - выкл, 1 - вкл	0	sync	u8 u8 svld 0 синхронизация работает
таймаут (мс)	500 500	wait	ul32 i32 sst 0 статус сервера
локальный IP адрес "[SQL]"	192.168.1.14	lip	str u8 ssl 255 service level сервера
локальный порт	8000 8000	lprt	i16 u32 sid 1 ID подписки
выбор сетевой платы / или /sock2 "/"	/	sdr	str u32 ssn 9571 номер уведомления подписки
IP адрес сервера "[SQL]"	192.168.1.12	rip	str u32 rcnt 9617 принято
порт сервера	4841 4841	rprr	i16 u32 wcnt 9618 отправлено
логин "admin"	admin	usr	str u8 prio 30 приоритет дополнительного потока (0 - отключен)
пароль "1"	1	psw	str u32 dsz 7 количество данных
приоритет потока (1-200), 0 - отключить	30 30	prio	u8 flt wrk 0.196583 текущее время работы (мс)
сброс временных счетчиков	0 0	rst	u8 flt mwrk 120.934 максимальное время работы (мс)
			flt scan 20.33 текущее время получения данных (мс)
			flt mscan 41.6684 максимальное время получения данных (мс)
			u16 diag 21286 диагностический счетчик

lip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_1"</sql>" (Пользовательское свойство 01)

rip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_4"</sql>" (Пользовательское свойство 04)

SyncMan 2		10	
SyncMan			1мкс
связь: 0 - выкл, 1 - вкл	1	enb	u8 i32 sts 1 0 - нет обмена, 1 - обмен
синхронизация: 0 - выкл, 1 - вкл	0	sync	u8 u8 svld 0 синхронизация работает
таймаут (мс)	500 500	wait	ul32 i32 sst 0 статус сервера
локальный IP адрес "[SQL]"	192.168.10.14	lip	str u8 ssl 255 service level сервера
локальный порт	8001 8001	lprt	i16 u32 sid 1 ID подписки
выбор сетевой платы / или /sock2 "/"	/	sdr	str u32 ssn 9615 номер уведомления подписки
IP адрес сервера "[SQL]"	192.168.10.12	rip	str u32 rcnt 9660 принято
порт сервера	4842 4842	rprr	i16 u32 wcnt 9662 отправлено
логин "admin"	admin	usr	str u8 prio 30 приоритет дополнительного потока (0 - отключен)
пароль "1"	1	psw	str u32 dsz 7 количество данных
приоритет потока (1-200), 0 - отключить	30 30	prio	u8 flt wrk 0.292834 текущее время работы (мс)
сброс временных счетчиков	0 0	rst	u8 flt mwrk 121.079 максимальное время работы (мс)
			flt scan 10.1445 текущее время получения данных (мс)
			flt mscan 42.8199 максимальное время получения данных (мс)
			u16 diag 21274 диагностический счетчик

Рисунок 5.26 – Успешный обмен: OPC UA-клиенты SyncMan (ПЛК2)

ip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_0"</sql>" (Пользовательское свойство 00)

OPC UA-сервер 1		5	
OpcUAServer			49мкс
192.168.1.X/24 "[SQL]"	192.168.1.14	ip	str i32 st 0 0 - running, 1 - failed (см. OPCUA Part5 12.6)
порт	4842 4842	prt	i16 uch cn 1 активных соединений
выбор сетевой платы / или /sock2 "/"	/	sdr	str uch cnDbg 0 соединений с отладчиком
0 - running, 1 - failed (см. OPCUA Part5 12.6)	0 0	st	i32 uch cnPA 1 соединений с клиентом PA
255 - ведущий, 199 - ведомый (см. OPCUA Part5 6.6.2.4.2)	255 255	sl	uch uch cnOth 0 соединений с другими клиентами
макс. кол-во соединений	255 255	max	uch

ip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_1"</sql>" (Пользовательское свойство 01)

OPC UA-сервер 2		10	
OpcUAServer			44мкс
192.168.10.X/24 "[SQL]"	192.168.10.14	ip	str i32 st 0 0 - running, 1 - failed (см. OPCUA Part5 12.6)
порт	4841 4841	prt	i16 uch cn 1 активных соединений
выбор сетевой платы / или /sock2 "/"	/	sdr	str uch cnDbg 0 соединений с отладчиком
0 - running, 1 - failed (см. OPCUA Part5 12.6)	0 0	st	i32 uch cnPA 1 соединений с клиентом PA
255 - ведущий, 199 - ведомый (см. OPCUA Part5 6.6.2.4.2)	255 255	sl	uch uch cnOth 0 соединений с другими клиентами
макс. кол-во соединений	255 255	max	uch

Рисунок 5.27 – Успешный обмен: OPC UA-серверы (ПЛК2)

lip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_0"</sql>" (Пользовательское свойство 00)
 rip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_3"</sql>" (Пользовательское свойство 03)

SyncMan 1		5		
SyncMan		2мкс		
связь: 0 - выкл, 1 - вкл	1 1	enb	u8 i32 sts 1 0 - нет обмена, 1 - обмен	2 - в резерве, >2 - переходное состояние, <0 -
синхронизация: 0 - выкл, 1 - вкл	0 0	sync	u8 u8 svld 0	синхронизация работает
таймаут (мс)	500 500	wait	ul32 i32 sst 0	статус сервера
локальный IP адрес "[SQL]"	192.168.1.12	lip	str u8 ssl 255	service level сервера
локальный порт	8000 8000	lprt	i16 u32 sid 1	ID подписки
выбор сетевой платы / или /sock2 "/"	/ /	sdr	str u32 ssn 17184	номер уведомления подписки
IP адрес сервера "[SQL]"	192.168.1.14	rip	str u32 rcnt 17262	принято
порт сервера	4841 4841	rprrt	i16 u32 wcnt 17263	отправлено
логин "admin"	admin	usr	str u8 prio 30	приоритет дополнительного потока (0 - отключен)
пароль "1"	1	psw	str u32 dsz 7	количество данных
приоритет потока (1-200), 0 - отключить	30 30	prio	u8 flt wrk 0.217584	текущее время работы (мс)
сброс временных счетчиков	0 0	rst	u8 flt mwrk 121.078	максимальное время работы (мс)
			flr scan 20.328	текущее время получения данных (мс)
			flr mscan 120.776	максимальное время получения данных (мс)
			u16 diag 34252	диагностический счетчик

lip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_1"</sql>" (Пользовательское свойство 01)
 rip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_4"</sql>" (Пользовательское свойство 04)

SyncMan 2		10		
SyncMan		1мкс		
связь: 0 - выкл, 1 - вкл	1 1	enb	u8 i32 sts 1 0 - нет обмена, 1 - обмен	2 - в резерве, >2 - переходное состояние, <0 -
синхронизация: 0 - выкл, 1 - вкл	0 0	sync	u8 u8 svld 0	синхронизация работает
таймаут (мс)	500 500	wait	ul32 i32 sst 0	статус сервера
локальный IP адрес "[SQL]"	192.168.10.12	lip	str u8 ssl 255	service level сервера
локальный порт	8001 8001	lprt	i16 u32 sid 1	ID подписки
выбор сетевой платы / или /sock2 "/"	/ /	sdr	str u32 ssn 17076	номер уведомления подписки
IP адрес сервера "[SQL]"	192.168.10.14	rip	str u32 rcnt 17151	принято
порт сервера	4842 4842	rprrt	i16 u32 wcnt 17155	отправлено
логин "admin"	admin	usr	str u8 prio 30	приоритет дополнительного потока (0 - отключен)
пароль "1"	1	psw	str u32 dsz 7	количество данных
приоритет потока (1-200), 0 - отключить	30 30	prio	u8 flt wrk 0.179667	текущее время работы (мс)
сброс временных счетчиков	0 0	rst	u8 flt mwrk 120.49	максимальное время работы (мс)
			flr scan 20.3321	текущее время получения данных (мс)
			flr mscan 100.281	максимальное время получения данных (мс)
			u16 diag 34245	диагностический счетчик

Рисунок 5.28 – Успешный обмен: OPC UA-клиенты SyncMan (ПЛК1)

Убедившись, что настройка обмена произведена успешно, можно приступить к настройке блока выбора текущих ролей контроллеров MasterSel. Для этого следует:

- Добавить блок **MasterSel** в проект.
Выход **L_Master** определяет роль контроллера: **L_Master = 0** – контроллер ведомый, **L_Master = 1** – контроллер ведущий.
- Так как сигнал с выхода **L_Master** может быть использован много раз в проекте, для удобства назначить его глобальной константой. Для этого в свойствах выхода **L_Master** следует добавить свойства **Полный алиас** и **Глобальная константа**.
- Вход блока **ready** (готовность) можно сразу задать **1** или завести на него другие сигналы из программы.

MasterSel		5		
MasterSel				
это ПЛК1	0	me1	b b L_Master	L_Master Ведущий
готовность	1	ready	b flt L_pila	Диагностический сигнал
инициализация	1	init	b b L_ready	Готов
команда Мастер1	0	master1	b b L_init_over	Иниц. завершена
команда Мастер2	0	master2	b b conn_fault	Нет связи с соседним ПЛК
таймер залипания пилы (мс)	200	tpila	l32 flt R_pila_1	Соседний ПЛК: диагн.сигнал 1
таймер восстановления связи (мс)	5000	trecon	l32 flt R_pila_2	Соседний ПЛК: диагн.сигнал 2
таймер на инициализацию (мс)	5000	tinit	l32 b R_ready	Соседний ПЛК Готов
			b R_master	Соседний ПЛК Мастер

Рисунок 5.29 – Настройка MasterSel

- Провести скрытую связь между выходом **L_Master** и входами **sync** блоков **SyncMan** через блок логического НЕ – **NOT** из библиотеки **paCore**, раздел **Логические**, так как синхронизацию следует включать на ведомом контроллере, когда **L_Master = 0**.

Провести скрытую связь можно кликая ПКМ на входе и выбирая в контекстном меню **Задать константу**.

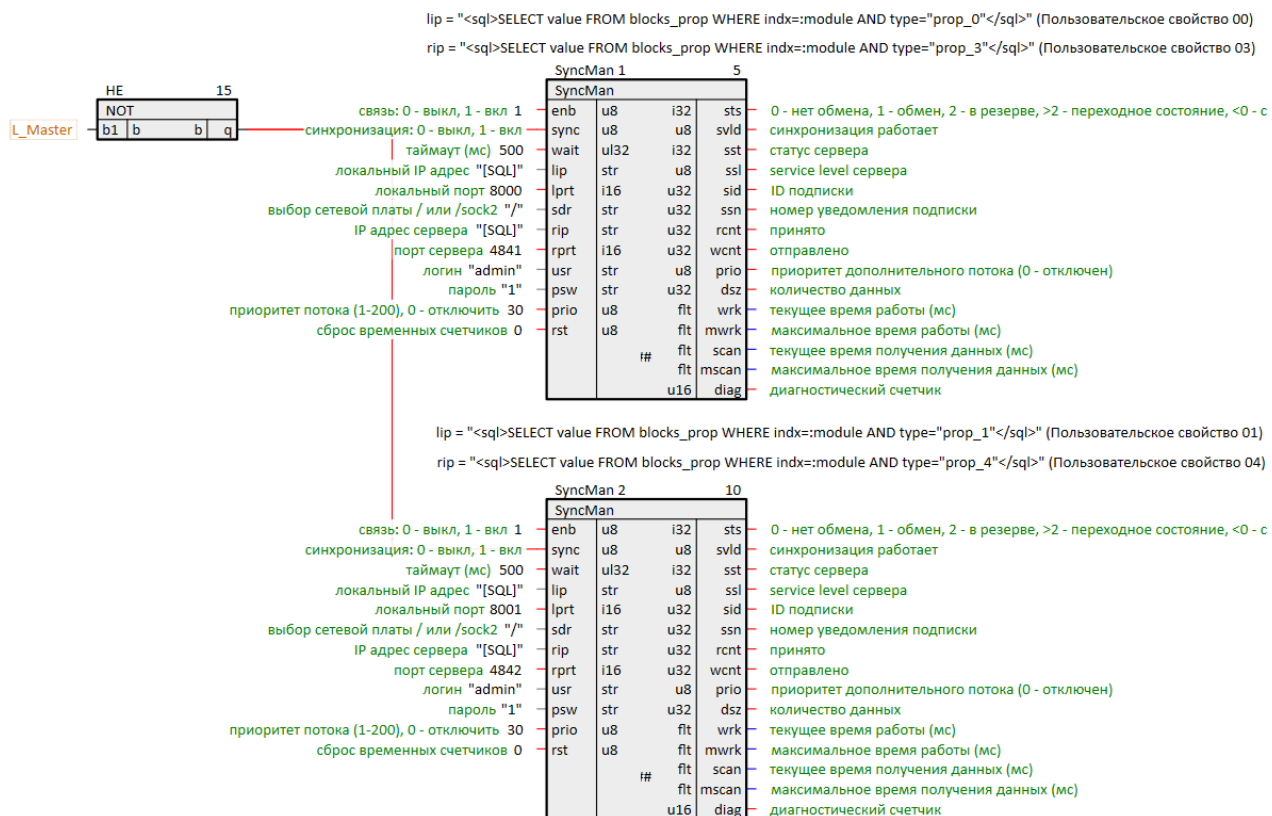


Рисунок 5.30 – Соединение MasterSel с SyncMan

- Запустить программу на первом контроллере. Он станет **ведущим** по истечении времени инициализации **tinit**. На выход **L_pila** выводится диагностический сигнал контроллера. Выход **conn_fault = 1**, так как программа на втором контроллере еще не включена.

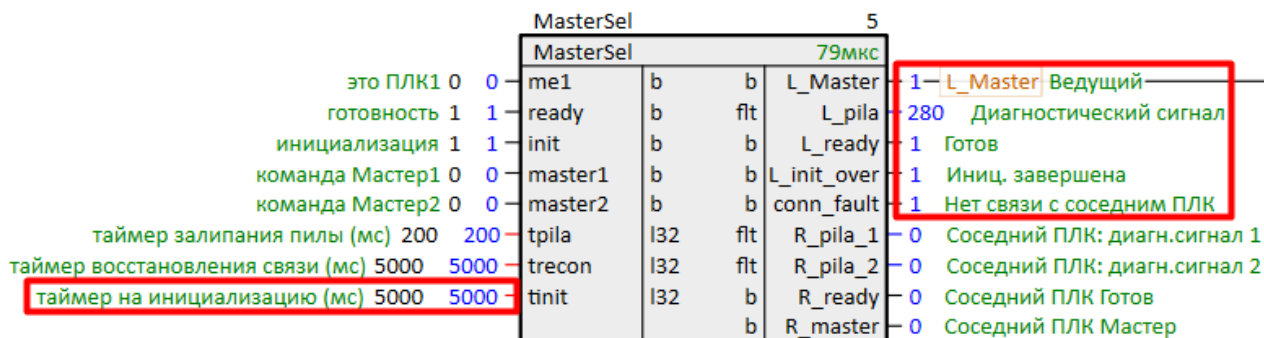


Рисунок 5.31 – Работа MasterSel (ПЛК1)

Установить **me1 = 1** – данный контроллер имеет признак **ПЛК1**, он будет становиться ведущим при неопределенных условиях. На данный вход можно завести сигнал с внешней кнопки или с панели оператора и т.п.



ПРИМЕЧАНИЕ

На входе **me1** можно задать константу при помощи SQL-запроса к пользовательскому свойству модуля и таким образом определить какой ПЛК является главным. Примеры запросов к свойствам модуля см. в [разделе 3](#).

MasterSel		5		
MasterSel		71мкс		
это ПЛК1	0 1	me1	b b L_Master 1	L_Master Ведущий
готовность	1 1	ready	b flt L_pila	1920 Диагностический сигнал
инициализация	1 1	init	b b L_ready	1 Готов
команда Мастер1	0 0	master1	b b L_init_over	1 Иниц. завершена
команда Мастер2	0 0	master2	b b conn_fault	1 Нет связи с соседним ПЛК
таймер залипания пилы (мс)	200 200	tpila	l32 flt R_pila_1	0 Соседний ПЛК: диагн.сигнал 1
таймер восстановления связи (мс)	5000 5000	trecon	l32 flt R_pila_2	0 Соседний ПЛК: диагн.сигнал 2
таймер на инициализацию (мс)	5000 5000	tinit	l32 b R_ready	0 Соседний ПЛК Готов
			b R_master	0 Соседний ПЛК Мастер

Рисунок 5.32 – Работа MasterSel (ПЛК1): задание признака ПЛК1

Запустить программу на втором контроллере. Он станет **ведомым** по истечении времени инициализации **tinit**, так как он имеет признак **ПЛК2**, а соседний контроллер имеет признак **ПЛК1**.

MasterSel		5		
MasterSel		77мкс		
это ПЛК1	0 0	me1	b b L_Master	0 L_Master Ведущий
готовность	1 1	ready	b flt L_pila	2960 Диагностический сигнал
инициализация	1 1	init	b b L_ready	1 Готов
команда Мастер1	0 0	master1	b b L_init_over	1 Иниц. завершена
команда Мастер2	0 0	master2	b b conn_fault	0 Нет связи с соседним ПЛК
таймер залипания пилы (мс)	200 200	tpila	l32 flt R_pila_1	800.001 Соседний ПЛК: диагн.сигнал 1
таймер восстановления связи (мс)	5000 5000	trecon	l32 flt R_pila_2	840.001 Соседний ПЛК: диагн.сигнал 2
таймер на инициализацию (мс)	5000 5000	tinit	l32 b R_ready	1 Соседний ПЛК Готов
			b R_master	1 Соседний ПЛК Мастер

Рисунок 5.33 – Работа MasterSel (ПЛК2)

На выходах **R_pila_1** и **R_pila_2** блока **MasterSel** у обоих контроллеров отобразится диагностический сигнал от соседнего контроллера. На выходе **R_ready** появится **1** – оба контроллера видят, что сосед готов.

На выходе **R_Master** ведомого контроллера появится **1** – **ПЛК2** видит, что сосед – ведущий контроллер.

MasterSel		5		
MasterSel		75мкс		
это ПЛК1	0 1	me1	b b L_Master	1 L_Master Ведущий
готовность	1 1	ready	b flt L_pila	8160 Диагностический сигнал
инициализация	1 1	init	b b L_ready	1 Готов
команда Мастер1	0 0	master1	b b L_init_over	1 Иниц. завершена
команда Мастер2	0 0	master2	b b conn_fault	0 Нет связи с соседним ПЛК
таймер залипания пилы (мс)	200 200	tpila	l32 flt R_pila_1	5880 Соседний ПЛК: диагн.сигнал 1
таймер восстановления связи (мс)	5000 5000	trecon	l32 flt R_pila_2	5840 Соседний ПЛК: диагн.сигнал 2
таймер на инициализацию (мс)	5000 5000	tinit	l32 b R_ready	1 Соседний ПЛК Готов
			b R_master	0 Соседний ПЛК Мастер

Рисунок 5.34 – Работа MasterSel (ПЛК1)

MasterSel		5		
MasterSel		78мкс		
это ПЛК1	0 0	me1	b b L_Master	0 L_Master Ведущий
готовность	1 1	ready	b flt L_pila	5880 Диагностический сигнал
инициализация	1 1	init	b b L_ready	1 Готов
команда Мастер1	0 0	master1	b b L_init_over	1 Иниц. завершена
команда Мастер2	0 0	master2	b b conn_fault	0 Нет связи с соседним ПЛК
таймер залипания пилы (мс)	200 200	tpila	l32 flt R_pila_1	3080 Соседний ПЛК: диагн.сигнал 1
таймер восстановления связи (мс)	5000 5000	trecon	l32 flt R_pila_2	3120 Соседний ПЛК: диагн.сигнал 2
таймер на инициализацию (мс)	5000 5000	tinit	l32 b R_ready	1 Соседний ПЛК Готов
			b R_master	1 Соседний ПЛК Мастер

Рисунок 5.35 – Работа MasterSel (ПЛК2)

Входы **master1** и **master2** блока **MasterSel** отвечают за ручную смену ролей контроллеров. На них можно завести сигналы с внешних кнопок или с панели оператора и т.п.

Для того чтобы передать роль ведущего от ПЛК1 к ПЛК2 следует подать **1** на вход **master2**.

MasterSel		5	
MasterSel		78мкс	
это ПЛК1 0 1	me1	b b	L_Master 0 L_Master Ведущий
готовность 1 1	ready	b flt	L_pila 2080 Диагностический сигнал
инициализация 1 1	init	b b	L_ready 1 Готов
команда Мастер1 0 0	master1	b b	L_init_over 1 Иниц. завершена
команда Мастер2 0 1	master2	b b	conn_fault 0 Нет связи с соседним ПЛК
таймер залипания пилы (мс) 200 200	tpila	l32 flt	R_pila_1 5520 Соседний ПЛК: диагн.сигнал 1
таймер восстановления связи (мс) 5000 5000	trecon	l32 flt	R_pila_2 5520 Соседний ПЛК: диагн.сигнал 2
таймер на инициализацию (мс) 5000 5000	tinit	l32 b	R_ready 1 Соседний ПЛК Готов
		b	R_master 1 Соседний ПЛК Мастер

Рисунок 5.36 – Работа MasterSel (ПЛК1): передача роли ведущего контроллера

ПЛК2 станет ведущим:

MasterSel		5	
MasterSel		74мкс	
это ПЛК1 0 0	me1	b b	L_Master 1 L_Master Ведущий
готовность 1 1	ready	b flt	L_pila 4240 Диагностический сигнал
инициализация 1 1	init	b b	L_ready 1 Готов
команда Мастер1 0 0	master1	b b	L_init_over 1 Иниц. завершена
команда Мастер2 0 0	master2	b b	conn_fault 0 Нет связи с соседним ПЛК
таймер залипания пилы (мс) 200 200	tpila	l32 flt	R_pila_1 520.001 Соседний ПЛК: диагн.сигнал 1
таймер восстановления связи (мс) 5000 5000	trecon	l32 flt	R_pila_2 560.001 Соседний ПЛК: диагн.сигнал 2
таймер на инициализацию (мс) 5000 5000	tinit	l32 b	R_ready 1 Соседний ПЛК Готов
		b	R_master 0 Соседний ПЛК Мастер

Рисунок 5.37 – Работа MasterSel (ПЛК2): получение роли ведущего контроллера

Условия автоматической смены ролей прописаны в [разделе 2.2](#) и в справке среды на блок **MasterSel**.



ПРИМЕЧАНИЕ

Блок **MasterSel** является составным, поэтому подробно логику его работы можно посмотреть на внутренних страницах. Для этого следует открыть библиотеку **paSync** в представлении **Дерево**.

Выход **L_Master** блока **MasterSel** можно вывести на внешние сигнальные лампы, панель оператора и т.д.

Рассмотрим организацию обмена с модулями линейки **Mx210**. В данном примере используются модули **МК210-311** (схема подключения на [рисунке 5.20](#)).

Для настройки TCP/IP-сервера в соответствии с [таблицей 5.2](#) следует использовать SQL-запрос к свойству модуля **Пользовательское свойство 02 (P1)**.

Настроить обмен с модулями по **Modbus TCP**.

Подробно настройка обмена по протоколу **Modbus** в среде Полигон рассмотрена в документе [Обмен по протоколу Modbus](#). Библиотека **paModbus**.

lip = "<sql>SELECT value FROM blocks_prop WHERE indx=:module AND type="prop_2"</sql>" (Пользовательское свойство 02)

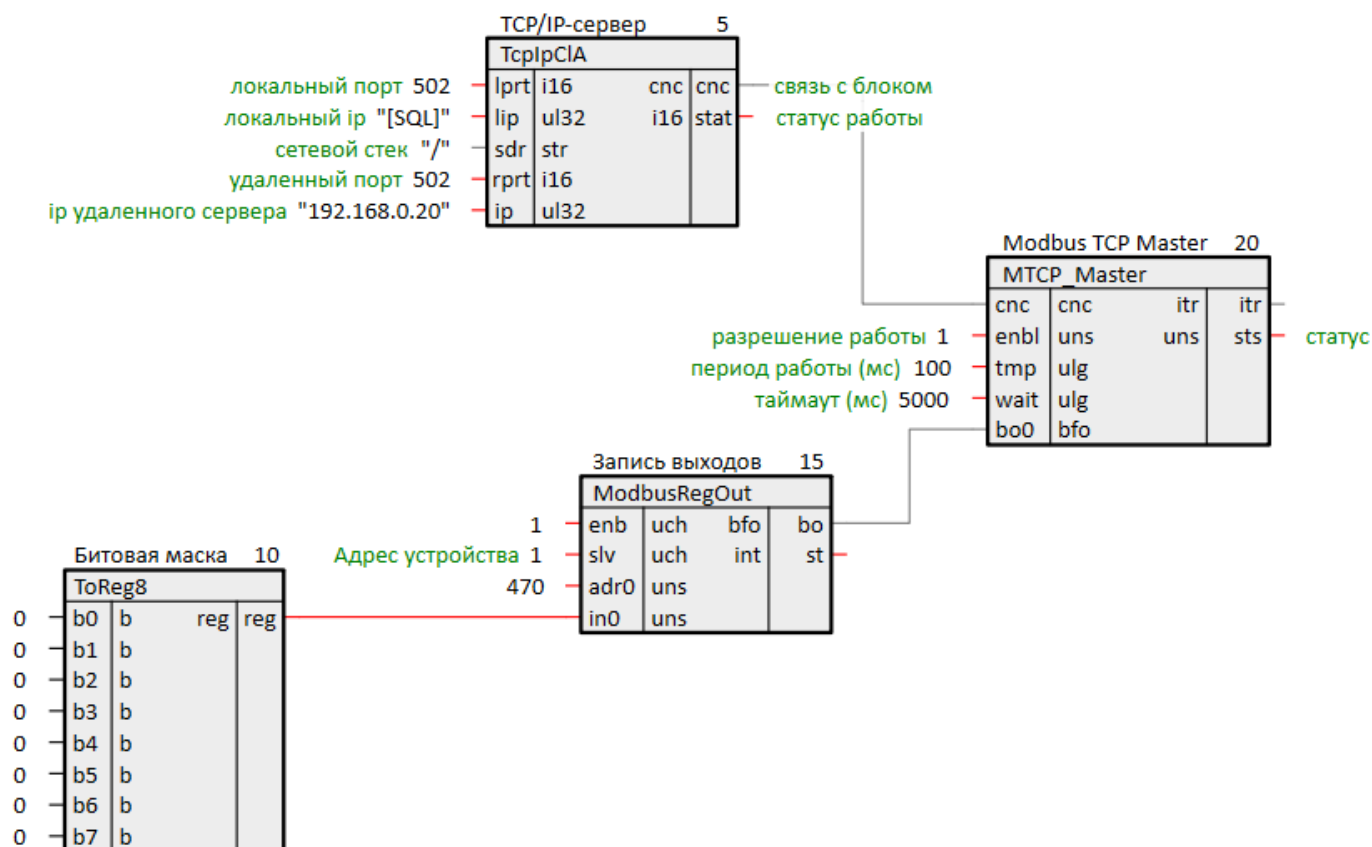


Рисунок 5.38 – Настройка обмена с модулем MK210-311

При данной настройке каждый контроллер будет опрашивать свой модуль MK210-311.



ПРИМЕЧАНИЕ

Пользователь может также настроить обмен с дублированными модулями в корзине, используя программный и/или внешний аппаратный решатель (арбитр) для определения достоверного сигнала – реализация резервирования корзины модулей Mx210.



Россия, 111024, Москва, 2-я ул. Энтузиастов, д. 5, корп. 5
тел.: +7 (495) 641-11-56, факс: (495) 728-41-45
тех. поддержка 24/7: 8-800-775-63-83, support@owen.ru
отдел продаж: sales@owen.ru
Веб-сайт ООО "ПромАвтоматика-Софт": www.pa.ru
рег.:1-RU-139911-1.1