

НАСТРОЙКА И РАБОТА REGUL OPC DA SERVER

Руководство пользователя

DPA-302.5

Версия 1.0.33

Февраль 2019

СОДЕРЖАНИЕ

Содержание	2
Назначение и возможности системы	3
Установка программы	4
Настройка конфигурации.....	7
Настройка среды Epsilon LD	10
Описание работы.....	12
Поддержка резервирования	16
Приложение А. Алгоритм выбора активного контроллера	17
Приложение Б. Набор тестов для проверки работы логики OPC-сервера	18
Приложение В. Ограничение пропускной способности	26
Приложение Г. Описание файла конфигурации	28

НАЗНАЧЕНИЕ И ВОЗМОЖНОСТИ СИСТЕМЫ

OPC сервер предназначен для доступа к экспортируемым переменным (символьной конфигурации проекта) приложения Epsilon LD через интерфейс OPC DA. Приложение служит для работы с контроллерами серии Regul RX00 как в одиночном исполнении, так и в составе резервированной системы.

По отношению к контроллерам OPC сервер обеспечивает:

- подключение по заданным IP-адресам к двум или одному (схема с резервированием и без) контроллерам Regul; по одному или двум сетевым интерфейсам к каждому контроллеру;
- доступ к экспортируемым переменным Epsilon LD проекта (символьной конфигурации) для чтения/записи произвольных наборов данных как с активного контроллера (в схеме с резервированием), так и с каждого контроллера отдельно.

Regul OPC DA Server – это:

- поддержка интерфейсов OPC DA 2.05a¹;
- поддержка работы с множественными подключениями OPC-клиентов;
- возможность назначения OPC-групп для работы как с тегами активного контроллера, так и для работы отдельно с каждым контроллером напрямую;
- поддержка переменных всех простых типов ИЕС, включая одномерные массивы простых типов (только чтение)²;
- поддержка VBA-клиентов (синхронное чтение/запись).

В системе резервирования OPC-сервер обеспечивает:

- четкий и однозначный алгоритм выбора активного контроллера (см. Приложение А);
- задержку переключения на резервный канал (интерфейс) при наличии резервного подключения (интерфейса) с момента обнаружения обрыва связи по активному каналу – примерно один период опроса данных с контроллера.

¹ Со стороны OPC-клиентов рекомендуются к использованию синхронный (IOPCSyncIO) и асинхронный (IOPCAsyncIO2) интерфейсы. «Старый» асинхронный интерфейс (IOPCAsyncIO) не поддерживается.

² Не поддерживаются или поддерживаются ограниченно (работа с корневыми элементами): двумерные массивы простых типов, массивы массивов простых типов, массивы структур и другие сложные типы данных.

УСТАНОВКА ПРОГРАММЫ

Для корректной работы программы перед установкой необходимо удалить все ранее установленные версии программы с компьютера. Деинсталляция производится стандартными средствами операционной системы.

Получите от предприятия-изготовителя файл установки с именем **ps_regul_opcda_setup_vX.X.X.exe**, где vX.X.X– номер версии программного обеспечения, например, ps_regul_opcda_setup_v1.0.33. Запустите файл от имени администратора. Откроется основное диалоговое окно мастера установки (Рисунок 1). Нажмите кнопку *Далее*.

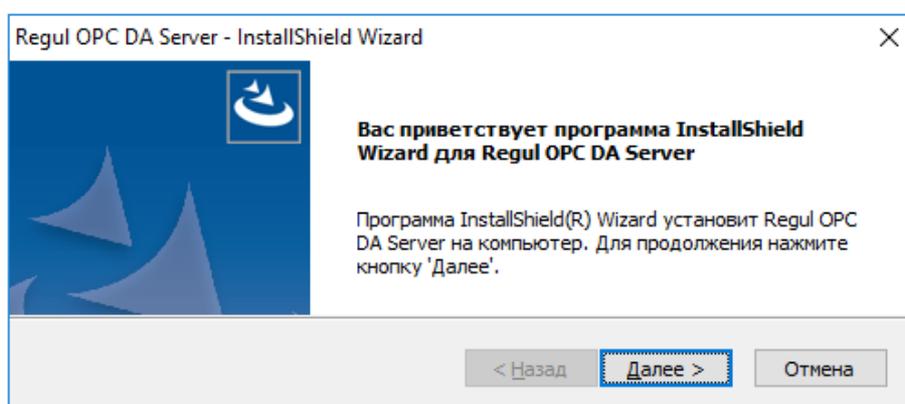


Рисунок 1. Диалоговое окно мастера установки Regul OPC DA Server

Откроется окно для выбора местоположения программы (Рисунок 2). По умолчанию программа установки создает папку **Prosoft-Systems** в каталоге **Program Files** (если операционная система Windows 32-bit) или в каталоге **Program Files (x86)** (если операционная система Windows 64-bit). Если требуется установить программу в другое место, то в поле **Конечная папка:** нажмите кнопку *Обзор...* и в открывшемся окне укажите, куда следует установить программу.

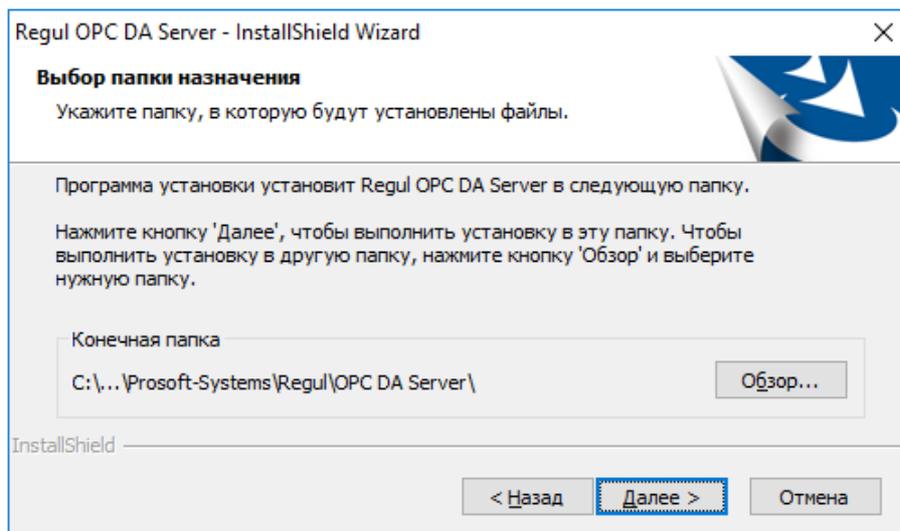


Рисунок 2. Диалоговое окно выбора папки, куда будет установлена программа

В окне **Выбор папки назначения** нажмите кнопку *Далее*. Откроется окно выбора компонентов программы для установки (Рисунок 3).

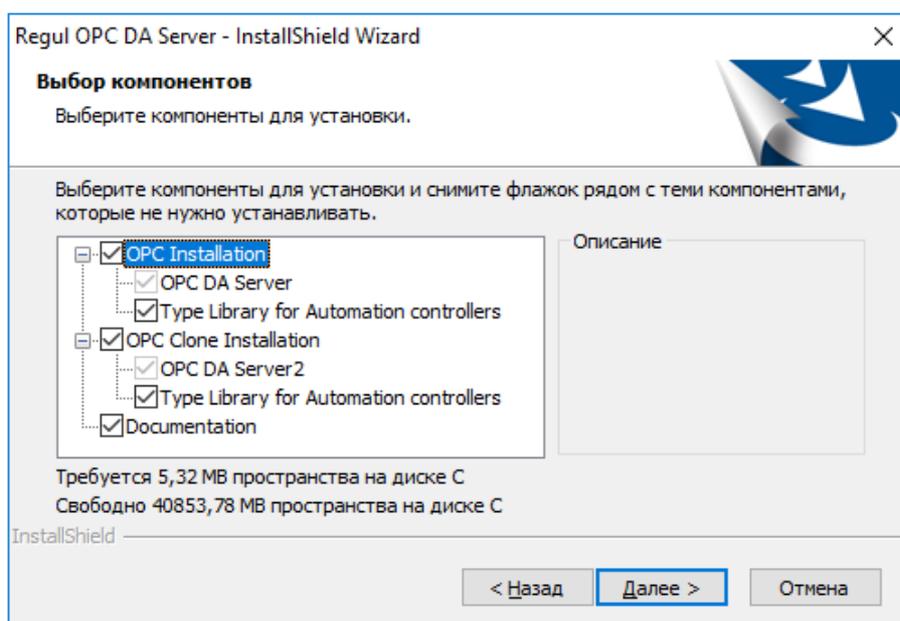


Рисунок 3. Диалоговое окно выбора компонентов для установки

Поставьте флажок возле нужных компонентов. Под перечнем компонентов в строке: «Требуется __ GB пространства на диске C» отображается объем свободного пространства жесткого диска, требуемый для корректной установки всех выбранных компонентов.

Имеется возможность установить максимум два независимых экземпляра OPC-сервера на одном компьютере (APMe) для работы с двумя парами контроллеров REGUL. В OPC-клиенте они доступны под именами **psregulopcda** и **psregulopcda2**.

Предлагаются к установке следующие компоненты:

- **OPC Installation** – установка OPC DA сервера Regul в составе OPC сервера и библиотеки автоматизации для работы с клиентами, поддерживающими VBA-скрипты.
- **OPC Clone Installation** – установка дополнительной независимой копии OPC DA сервера для возможности работы с двумя парами контроллеров Regul с одного АРМа. Также содержит отдельную библиотеку автоматизации для VBA-клиентов.
- **Documentation** – руководство пользователя по установке и настройке OPC DA сервера Regul.

После выбора компонентов для установки нажмите кнопку *Далее*. Откроется диалоговое окно с сообщением, что мастер готов выполнить установку Regul OPC DA Server. Чтобы начать установку нажмите кнопку *Установить*.

Начнется копирование файлов и установка программы, что может занять некоторое время. Дождитесь окончания процесса и появления оповещения об успешном окончании установки. Нажмите кнопку *Готово*. По окончании установки программы автоматически будет создан ярлык в меню **Пуск**.

В ходе установки исполняемый файл будет автоматически зарегистрирован как COM OPC DA сервер и станет доступен для подключения OPC-клиентам.

НАСТРОЙКА КОНФИГУРАЦИИ

После установки приложения конфигурационный файл **config.ini** располагается в папке:

c:\ProgramData\Prosoft-Systems\Regul\OPC DA Server\

Примечание: эта папка может быть скрытая.

Полное описание параметров, содержащихся в файле **config.ini**, приведено в таблице 7 приложения Г.

Наиболее общие и употребительные настройки вынесены в диалог настроек приложения (Рисунок 5), доступного через пункт основного меню **Tools** ⇨ **Settings...** (Рисунок 4).

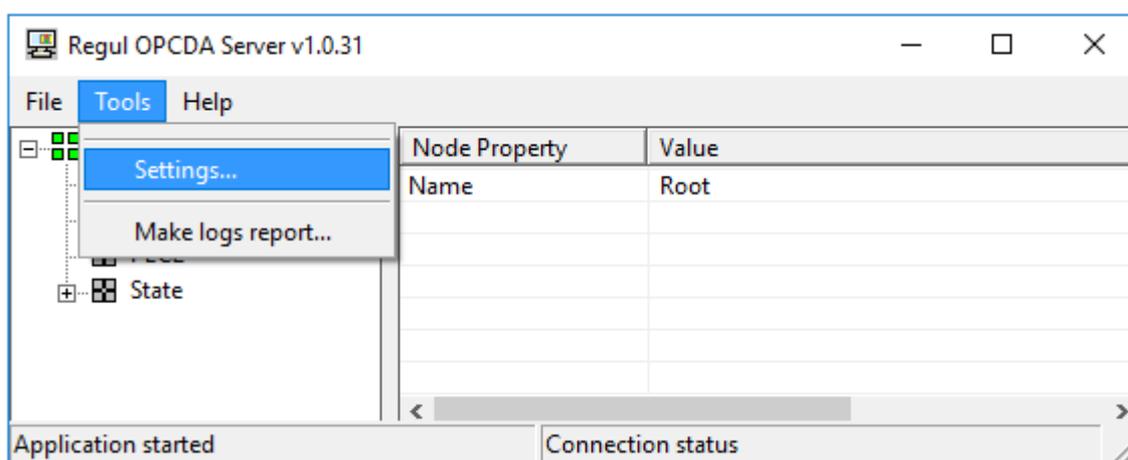


Рисунок 4. Пункт основного меню Tools ⇨ Settings

Пункт меню **Tools** ⇨ **Make logs report...** при возникновении нештатной ситуации в работе OPC-сервера позволяет автоматически сохранить в один архив все лог-файлы приложения и файл настроек для дальнейшей отправки разработчику (портал техподдержки Prosoft-Systems).

В окне **Settings** доступны несколько подгрупп настроек – Redundancy, PLC, OPC, Security, User Interface, связанные с соответствующими параметрами из конфигурационного файла.

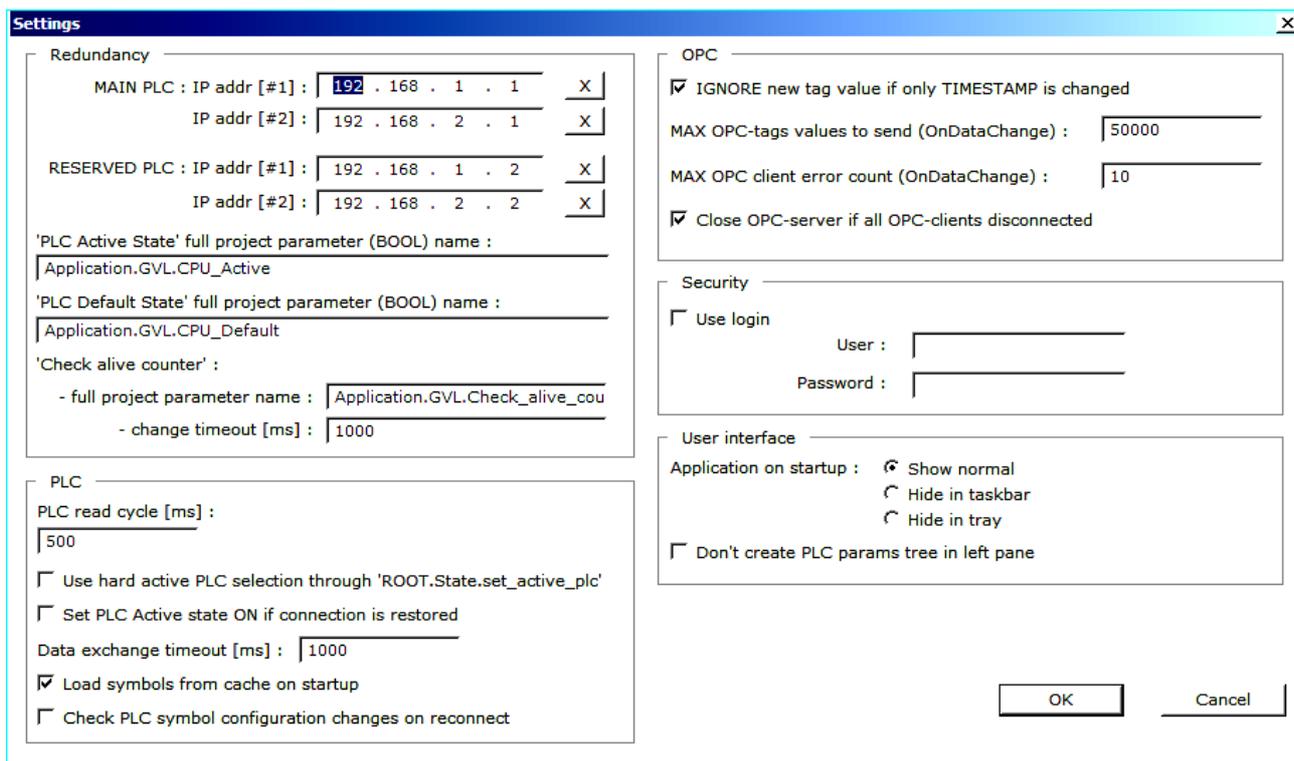


Рисунок 5. Основные настройки конфигурации

Для установки настроек руководствуйтесь описанием полей и параметров (Таблица 1, Таблица 7 приложения Г).

Таблица 1. Параметры конфигурации

№ п/п	Поле	Соответствующий параметр в файле конфигурации	Номер параметра в таблице 7
1	MAIN PLC :: IP addr [#1] IP addr [#2] RESERVED PLC :: IP addr [#1] IP addr [#2]	plc1_port1 plc1_port1 plc2_port1 plc2_port2	2.1
2	PLC read cycle [ms]	plc_params_check_period	2.13
3	Redundancy : 'PLC Active State'	plc_main_check_tag	2.6
4	Redundancy : 'PLC Default State'	plc_main_check_tag2	2.7
5	Check alive counter: param name	plc_check_alive_counter_tag	2.20
6	Check alive counter: timeout [ms]	plc_check_alive_counter_to	2.21
7	Use hard active PLC selection through 'ROOT.State.set_active_plc'	plc_use_hard_arm_selection	2.8

№ п/п	Поле	Соответствующий параметр в файле конфигурации	Номер параметра в таблице 7
8	Set PLC Active state ON if connection is restored	plc_set_active_on_link_restore	2.9
9	Data exchange timeout [ms]	plc_data_exchange_timeout	2.22
10	Load symbols from file on startup	plc_load_symbols_from_file	2.25
11	Check PLC symbol configuration changes on reconnect	plc_check_symbols_change	2.26
12	IGNORE new tag value if only TIMESTAMP is changed	update_tags_if_only_timestamp_changed	1.10, значение инвертировано
13	MAX OPC-tags values to send (OnDataChange)	max_opc_tags_to_send	1.7
14	MAX OPC client error count (OnDataChange)	opc_client_max_errors	1.9
15	Close OPC-server if all OPC-clients disconnected	close_on_last_opc_client	1.8
16	Use login	plc_use_user_login	2.2
17	Login	plc_user_login	2.3
18	Password	plc_user_password	2.4
19	Application on startup: <ul style="list-style-type: none"> – Show normal – Hide in taskbar – Hide in tray 	1.11 и 1.12 не заданы (= 0) on_startup_hide_in_taskbar on_startup_hide_in_tray	1.11 и 1.12 1.11 1.12
20	Don't create PLC params tree in left pane	dont_create_params_tree	1.13

Потребуется перезапуск приложения OPC-сервера при изменении следующих настроек:

- PLC :: IP addr...,
- Redundancy :: PLC Active State...,
- Redundancy :: PLC Default State...,
- Redundancy :: Check alive counter...,
- Use user login,
- Login,
- Password,
- Don't create PLC params tree in left pane.

НАСТРОЙКА СРЕДЫ EPSILON LD

Для того, чтобы данные с ПЛК передавались посредством протокола OPC, необходимо в программе Epsilon LD добавить компонент Symbol Configuration (символьная конфигурация) в опциях проекта IEC-приложения. В контекстном меню приложения (Application) выберите Добавить объект (Add object) и в контекстном меню выберите пункт Symbol Configuration... (Рисунок 6).

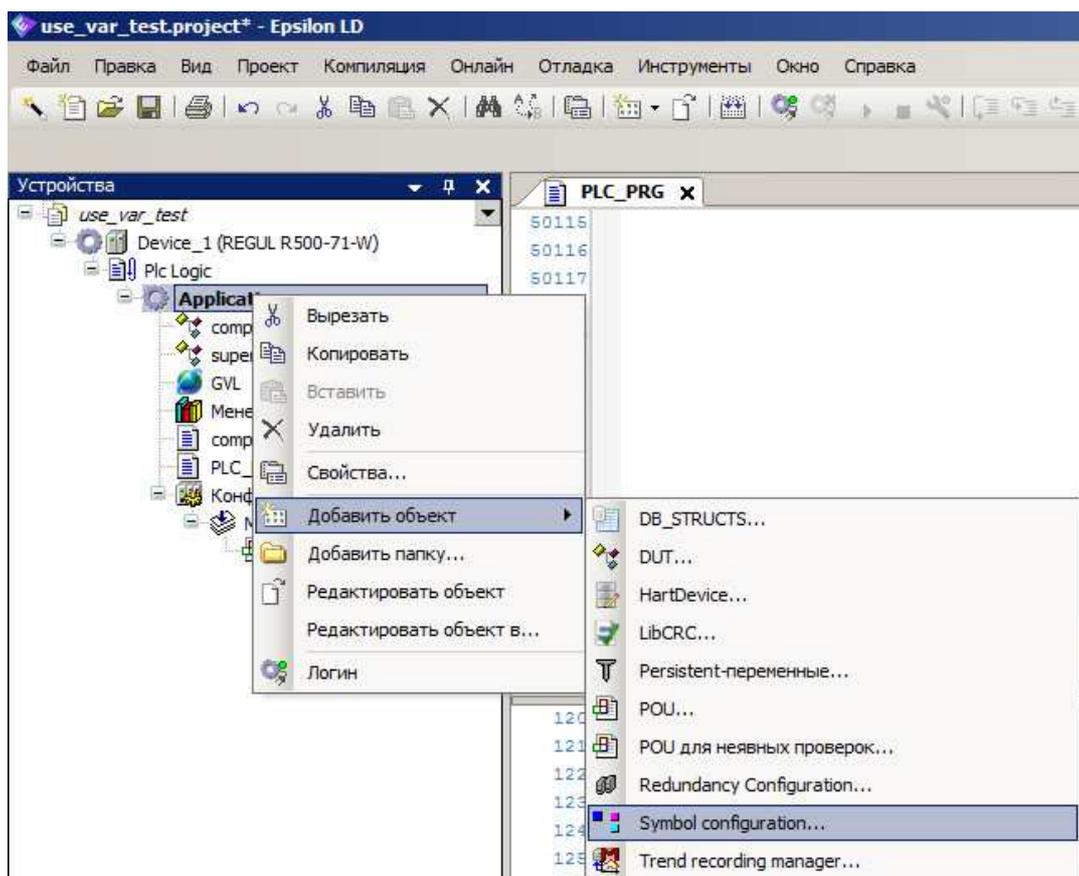


Рисунок 6. Добавление компонента символьной конфигурации

Для добавления переменных приложения выберите объект **Symbol Configuration** в дереве устройств (двойной щелчок мыши по названию объекта). Откроется вкладка **Symbol Configuration**. Выберите нужную папку (PLC_PRG, GVL и т.д.) и в ней появится список переменных, определенных в IEC-приложении (Рисунок 7). Установите флажок напротив тех переменных, взаимодействие с которыми будет обеспечиваться протоколом OPC.

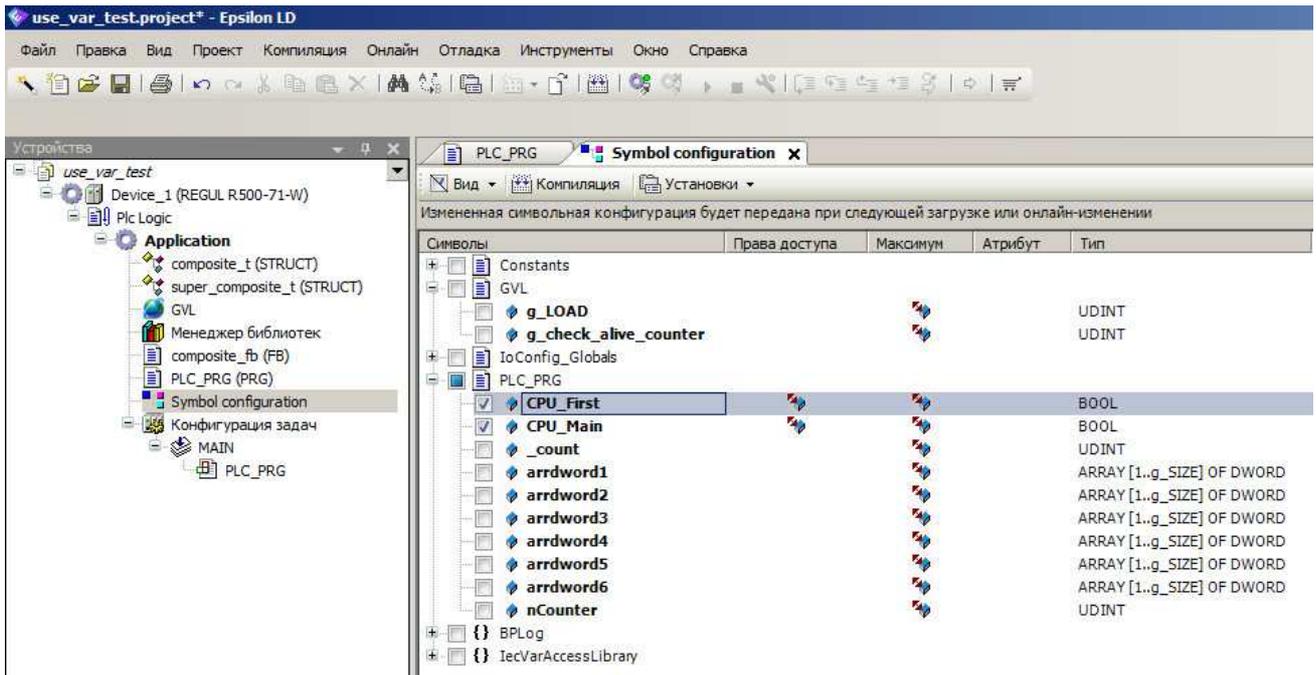


Рисунок 7. Выбор переменных проекта для экспорта

ОПИСАНИЕ РАБОТЫ

При подключении OPC-сервера к контроллерам хотя бы по одному каналу (здесь канал – это подключение по одному из IP-адресов к любому ПЛК), в левой части окна приложения в ветках **Root.PLC/PLC1/PLC2** отобразится дерево переменных символьной конфигурации проекта (см. раздел выше), загруженного в контроллер (Рисунок 8).

ВНИМАНИЕ! Во всех ветках набор переменных будет одинаков, так как предполагается, что OPC-сервер работает с двумя ПЛК в резерве, либо с двумя ПЛК, на которые загружен один и тот же проект.

TagName	Value	Time	Quality	VarType	VarAccess
active_plc	1	2019-01-16 12:42:01....	Good	VT_I4	R
link1_name	172.29.23.0	2019-01-16 12:42:00....	Good	VT_BSTR	R
link1_status	1	2019-01-16 12:42:00....	Good	VT_BOOL	R
link2_name		2019-01-16 12:42:00....	Good	VT_BSTR	R
link2_status	0	2019-01-16 12:42:00....	Good	VT_BOOL	R
plc_11	2	2019-01-16 12:42:01....	Good	VT_I4	R
plc_11_ip	172.29.23.215	2019-01-16 12:42:00....	Good	VT_BSTR	R
plc_12	0	2019-01-16 12:42:00....	Good	VT_I4	R
plc_12_ip		2019-01-16 12:42:00....	Good	VT_BSTR	R
plc_21	1	2019-01-16 12:42:02....	Good	VT_I4	R
plc_21_ip	172.29.23.216	2019-01-16 12:42:00....	Good	VT_BSTR	R
plc_22	0	2019-01-16 12:42:00....	Good	VT_I4	R
plc_22_ip		2019-01-16 12:42:00....	Good	VT_BSTR	R
plc_link_error	0	2019-01-16 12:42:00....	Good	VT_BOOL	R
set_active_plc	0	2019-01-16 12:42:00....	Good	VT_I4	R/W

Рисунок 8. Дерево переменных проекта

В ветке **Root.State** отображаются теги состояния подключения и сетевых интерфейсов.

Теги **Root.State.plc_xy** (x – номер ПЛК, y – номер интерфейса) содержат код состояния подключения к контроллеру:

- 0 – нет подключения;
- 1 – подключение установлено;
- 2 – подключение установлено и этот канал является активным на данный момент.

Теги **Root.State.plc_xy_ip** содержат IP-адреса каналов из конфигурационного файла.

Теги **Root.State.linkX_name** показывают номер подсети соответствующего сетевого интерфейса.

Теги **Root.State.linkX_status** – статус соответствующего сетевого интерфейса (сетевой кабель подключен/обрыв – 1/0).

Тег **Root.State.plc_link_error** принимает значение 1 в случае потери связи между контроллерами в схеме с резервированием (оба являются активными).

Тег **Root.State.set_active_plc** позволяет управлять выбором ведущего контроллера с верхнего уровня (АРМа) согласно алгоритму (см. Приложение А) и настроечным параметрам 2.8 и 2.9. Запись в тег значения 1 или 2 задает номер ведущего контроллера, прочие значения выключают управление с верхнего уровня.

В теге **Root.State.active_plc** отображается номер (1,2) текущего активного контроллера или 0 при его отсутствии.

Ветка **Root.PLC** является «общей» – в ней отображаются значения переменных, полученных по активному на данный момент каналу связи с активного (с точки зрения OPC-сервера) контроллера в схеме с резервированием (и без).

Дополнительные ветки **Root.PLC1** и **Root.PLC2** используются для обмена данными напрямую с каждым контроллером по отдельности.

ВНИМАНИЕ! В интерфейсе программы отображаются/обновляются значения только тех переменных, на которые есть подписка со стороны OPC-клиентов, и которые вычитываются с контроллеров.

Пояснение к п.2.1 настроек (таблица 7 приложения Г)

По умолчанию, **plc1_port1** и **plc2_port1** принадлежат к одной независимой подсети, а **plc1_port2** и **plc2_port2** – к другой. Например,

`plc1_port1 = 192.168.28.1/24`

`plc1_port2 = 192.168.34.1/24`

`plc2_port1 = 192.168.28.2/24`

`plc2_port2 = 192.168.34.2/24`

Допускается настройка подключения к одиночному контроллеру REGUL по одному или двум интерфейсам.

Пояснение к пп.2.6-2.7 настроек (таблица 7 приложения Г)

При работе с двумя ПЛК в резерве OPC-сервер определяет активный ПЛК на основании значений двух флагов – переменных, которые добавлены в символьную конфигурацию проекта и считываются с самих ПЛК. При наличии в проекте компонента **Резервирование / Redundancy** (см. документацию «Конфигурирование резервированной системы на контроллерах серии Regul RX00») указанные переменные, например, определяются следующим способом.

В одном из POU (например, *Redundancy_POU*) проекта задается следующий программный код (на языке ST):

Объявление

```
RedMode : SINT;
IsStateActive : BOOL; //флаг активности ПЛК
IsDefaultPlc : BOOL; //флаг ПЛК по умолчанию
```

Реализация

```
PsRedundancy.Synchronize(FALSE);
RedMode := PsRedundancy.GetMode();
IsStateActive := (RedMode=PsRedundancy.ACTIVE) OR (RedMode=PsRedundancy.ACTIVE_STANDALONE);
IsDefaultPlc := PsRedundancy.ThisIsDefaultPlc();
```

Соответственно, в настройках пп 2.6-2.7 задается полный путь к этим переменным в виде *Имя_Приложения_Проекта / Имя_POU / Имя_Переменной*, например, в нашем случае:

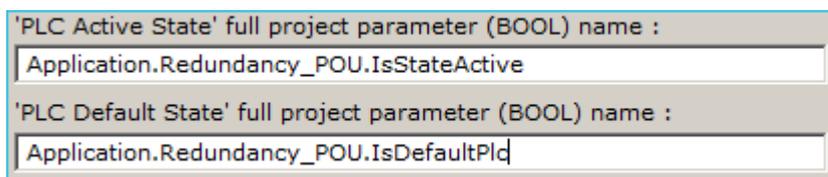


Рисунок 9. Задание флагов выбора активного ПЛК

ВАЖНО! В случае обрыва связи с ПЛК, когда OPC-сервер не может получить актуальные значения указанных флагов, выбор активного ПЛК с точки зрения OPC-сервера может не совпадать с фактическим состоянием подсистемы резервирования двух ПЛК.

Пояснение к пп.2.11-2.12 настроек (таблица 7 приложения Г)

По умолчанию, для работы напрямую с первым и вторым контроллером создаются ветки **Root.PLC1** и **Root.PLC2**. OPC-клиенты могут работать с тегами из этих групп для чтения и записи в определенный ПЛК. При OPC-браузинге путь к тегам показывается, начиная с префиксов **PLC1** и **PLC2**. Под «первым» и «вторым» ПЛК здесь понимаются контроллеры, для которых IP-адреса заданы в настройках *[plc1_port1, plc1_port2]* и *[plc2_port1, plc2_port2]* соответственно.

ВАЖНО! Для того, чтобы OPC-группа была привязана к конкретному контроллеру, OPC-клиент должен добавлять в эту OPC-группу только OPC-теги с соответствующим префиксом – либо **PLC1**, либо **PLC2**. Если, по ошибке, в одну OPC-группу будут добавлены OPC-теги с

разными префиксами, то OPC-группа будет привязана к конкретному ПЛК по префиксу последнего добавленного OPC-тега.

При этом запись в эти теги будет производиться в конкретный контроллер (при наличии с ним связи), независимо от того, является ли он активным на данный момент.

Чтение переменных контроллера (при наличии с ним связи), связанных с такими OPC-тегами, будет также производиться из этого контроллера независимо от того, является ли этот контроллер активным на данный момент.

При потере связи с контроллером у всех связанных с ним тегов в соответствующей подгруппе выставляется качество BAD.

Пояснение к п.1.13 настроек (таблица 7 приложения Г)

При подключении к ПЛК в окне приложения отображается дерево всех(!!!) переменных из символьной конфигурации в тройном объеме (ветки PLC1 и PLC2 являются дубликатами PLC). Если нет необходимости в наблюдении текущих значений переменных в окне приложения самого OPC-сервера, то рекомендуется отключить построение дерева переменных с помощью этой настройки. Иначе, при большом количестве экспортируемых в символьной конфигурации переменных проекта (от 10000 и более), построение дерева занимает продолжительное время и блокирует графический интерфейс пользователя. Обновление значений переменных внутреннего представления OPC-сервера при этом создает дополнительную нагрузку на процессор АРМа.

Другим вариантом является уменьшение отмеченных галочками переменных в символьной конфигурации проекта Epsilon LD до минимального необходимого объема.

Пояснение к п.2.26 настроек (таблица 7 приложения Г)

Возможно включение этой настройки на время разработки и отладки проекта, когда при заливке приложения в контроллер возможно изменение символьной конфигурации проекта и это требуется отслеживать без перезапуска OPC-сервера. Большое число переменных в символьной конфигурации (>10000) приводит к существенным задержкам на этапе сравнения старой и новой конфигурации, поэтому в рабочем режиме эту настройку рекомендуется отключать.

ПОДДЕРЖКА РЕЗЕРВИРОВАНИЯ

OPC-сервер Regul предлагает набор механизмов поддержки резервирования, позволяющих отслеживать активность одного из контроллеров и, соответственно, переключать передачу данных в OPC-интерфейс по выбранному каналу обмена данными с контроллером согласно следующим алгоритмам, выбранному типу резервирования и настройкам OPC-сервера.

Аппаратное резервирование

В случае использования аппаратного резервирования согласно документации «Конфигурирование резервированной системы на контроллерах серии Regul RX00» необходимо настроить п.п.2.6, 2.7 (см. таблицу 7 приложения Г), добавив соответствующие переменные в символьную конфигурацию проекта Epsilon LD. Выбор активного контроллера при этом будет синхронизирован с работой компонента Redundancy.

Программное резервирование

Выбор активного контроллера (из двух контроллеров с одинаковым проектом) выполняется на основании:

- наличия подключения к обоим контроллерам по двум интерфейсам;
- явного выбора активного контроллера через тег *Root.State.set_active_plc*;
- настроек п.п.2.5, 2.8, 2.9.

Вспомогательные механизмы

Контрольный счетчик – активируется в настройках п.п.2.20, 2.21. Служит для обработки внештатной ситуации (исключения) в работе приложения/среды исполнения, когда все значения переменных символьной конфигурации, включая флаги аппаратного резервирования, замораживаются (с достоверным качеством значения), а выполнение приложения (рабочего цикла) останавливается. Может применяться в обоих случаях резервирования. Выполняется на самом последнем этапе выбора активного контроллера.

Состояние приложения (RUN/STOP) на контроллере – настройка п.2.10. Рекомендуется использовать **в отладочных целях**, так как достоверный результат обеспечивается только если приложение на контроллере одно или все приложения на контроллере имеют одинаковый статус выполнения. Дополнительно, каждый запрос статуса приложения на ПЛК приводит к генерации системных событий Login / Logout, что может быть нежелательно для конечного пользователя.

ПРИЛОЖЕНИЕ А. АЛГОРИТМ ВЫБОРА АКТИВНОГО КОНТРОЛЛЕРА

При настроенном резервировании OPC-сервер ориентируется на переменные приложения контроллера. Эти переменные определяют контроллер, активный на данный момент (параметр 2.6 – `plc_main_check_tag = Application.GVL.CPU_Active`) и контроллер по умолчанию (параметр 2.7 – `plc_main_check_tag2= Application.GVL.CPU_Default`).

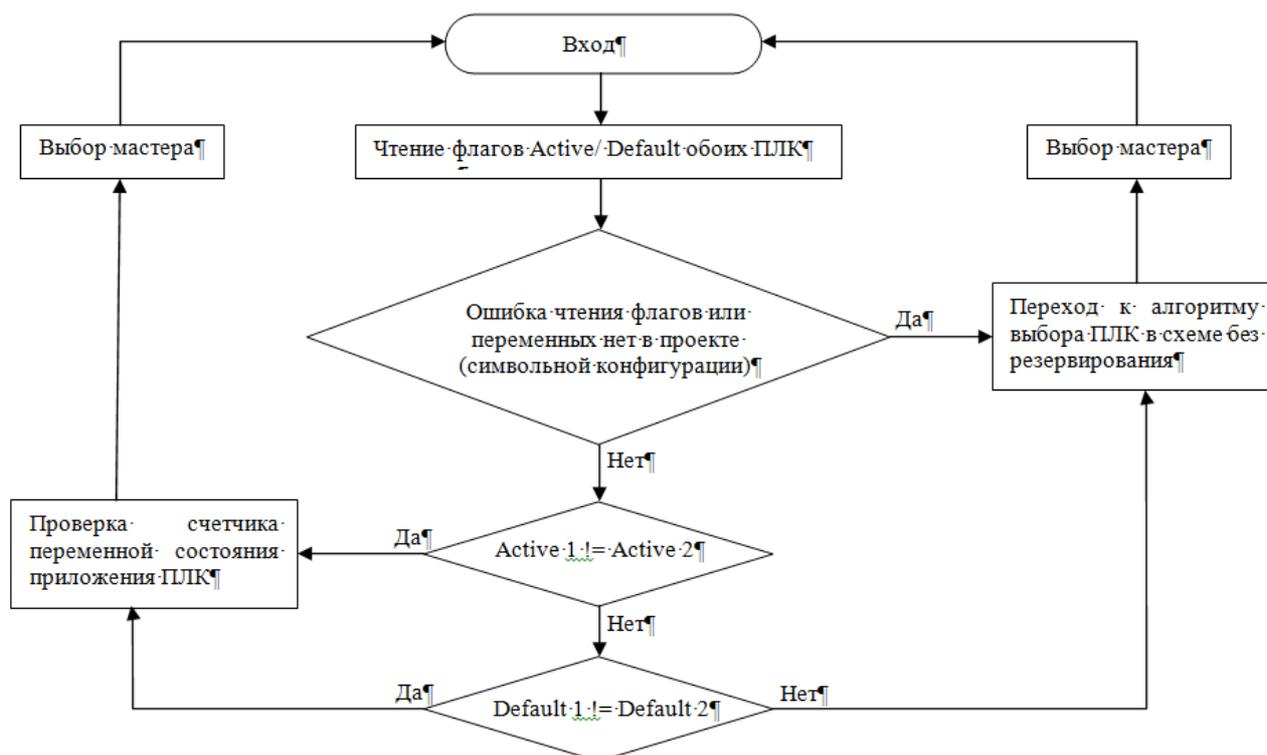


Рисунок 10. Алгоритм выбора активного контроллера

Иначе выбор происходит согласно алгоритму без резервирования. При этом анализируется:

- наличие подключения по четырем каналам к двум контроллерам;
- настройки параметров 2.5, 2.8, 2.9 (Таблица);
- запись в тег выбора активного контроллера `Root.State.set_active_plc`.

Если выполнена настройка параметра 2.20, то на последнем шаге после выбора активного контроллера дополнительно происходит проверка счетчика переменной отслеживания активности приложения/среды выполнения для этого контроллера и окончательный выбор, какой контроллер считать активным.

ПРИЛОЖЕНИЕ Б. НАБОР ТЕСТОВ ДЛЯ ПРОВЕРКИ РАБОТЫ ЛОГИКИ ОПС-СЕРВЕРА

Внимание! Тесты проводятся без настройки параметра 2.20 – `plc_check_alive_counter_tag = Application.GVL.Check_alive_counter`.

Два контроллера без резервирования с одинаковыми приложениями (проектами)

Связь между контроллерами и АРМ по одному или двум интерфейсам.

Дерево тегов ОПС-сервера:

- ROOT.PLC. - теги общей подгруппы активного на данный момент контроллера;
- ROOT.PLCx - подгруппа тегов для работы напрямую с каждым контроллером (x = 1,2);

В ОПС-клиенте создаются три ОПС-группы с одинаковым набором ОПС-тегов из трех подгрупп ROOT.PLC, ROOT.PLC1, ROOT.PLC2.

Настройки ОПС-сервера:

- `plc_use_hard_arm_selection = 0`;
- `plc_set_active_on_link_restore = 0`.

Таблица 2. Набор тестов

N	Название теста	Исходное состояние	Требуемое поведение
1	Проверка переключения при перезагрузке активного контроллера или обрыве с ним связи	ОПС-сервер подключен к обоим контроллерам; активен один из контроллеров, в общей подгруппе тегов ROOT.PLC отображаются значения с активного контроллера	активным становится второй контроллер; теги в общей подгруппе ROOT.PLC принимают значения со второго контроллера; теги в подгруппе первого контроллера ROOT.PLCx получают качество BAD
2	Проверка переключения при восстановлении связи с контроллером из теста 1	конечное состояние из теста 1	активным остается второй контроллер; теги в подгруппе первого контроллера ROOT.PLCx получают качество GOOD

N	Название теста	Исходное состояние	Требуемое поведение
3	Проверка переключения через тег ROOT.STATE.set_active_plc	ОПС-сервер подключен к обоим контроллерам; активен один из контроллеров, в общей подгруппе тегов ROOT.PLC. отображаются значения с активного контроллера	активным становится второй контроллер; теги в общей подгруппе ROOT.PLC принимают значения со второго контроллера
4	Проверка перезагрузки или обрыва связи с обоими контроллерами	ОПС-сервер подключен к обоим контроллерам; активен один из контроллеров, в общей подгруппе тегов ROOT.PLC отображаются значения с активного контроллера	теги во всех подгруппах получают качество BAD
5	Проверка восстановления связи с обоими контроллерами	конечное состояние из теста 4	исходное состояние из теста 4; теги во всех подгруппах получают качество GOOD
6	Проверка записи в теги контроллера	ОПС-сервер подключен к обоим контроллерам; активен один из контроллеров, в общей подгруппе тегов ROOT.PLC отображаются значения с активного контроллера	при записи в теги общей подгруппы ROOT.PLC или подгруппы ROOT.PLCx активного контроллера, значения меняются синхронно в обеих этих группах; при записи в теги подгруппы ROOT.PLCx неактивного контроллера, значения меняются только в этой группе

В тестах 1-5 все изменения статуса подключения ОПС-сервера к контроллерам должны отображаться и адекватно меняться в соответствующих тегах подгруппы ROOT.STATE согласно Руководству по эксплуатации.

Настройки ОПС-сервера:

- plc_use_hard_arm_selection = 0;
- plc_set_active_on_link_restore = 1.

Все пункты тестов аналогичны тестам таблицы 2 за исключением теста 2.

Таблица 3. Набор тестов

N	Название теста	Исходное состояние	Требуемое поведение
1	Проверка переключения при перезагрузке активного контроллера или обрыве с ним связи	<p>ОПС-сервер подключен к обоим контроллерам;</p> <p>активен один из контроллеров, в общей подгруппе тегов ROOT.PLC отображаются значения с активного контроллера</p>	<p>активным становится второй контроллер;</p> <p>теги в общей подгруппе ROOT.PLC принимают значения со второго контроллера;</p> <p>теги в подгруппе первого контроллера ROOT.PLCx получают качество BAD</p>
2	Проверка переключения при восстановлении связи с контроллером из теста 1	<p>конечное состояние из теста 1</p>	<p>активным снова становится первый контроллер;</p> <p>теги в подгруппе первого контроллера ROOT.PLCx получают качество GOOD;</p> <p>теги в общей подгруппе ROOT.PLC принимают значения с первого контроллера</p>
3	Проверка переключения через тег ROOT.STATE.set_active_plc	<p>ОПС-сервер подключен к обоим контроллерам;</p> <p>активен один из контроллеров, в общей подгруппе тегов ROOT.PLC отображаются значения с активного контроллера</p>	<p>активным становится второй контроллер;</p> <p>теги в общей подгруппе ROOT.PLC принимают значения со второго контроллера</p>
4	Проверка перезагрузки или обрыва связи с обоими контроллерами	<p>ОПС-сервер подключен к обоим контроллерам;</p> <p>активен один из контроллеров, в общей подгруппе тегов ROOT.PLC отображаются значения с активного контроллера</p>	<p>теги во всех подгруппах получают качество BAD</p>
5	Проверка восстановления связи с обоими контроллерами	<p>конечное состояние из теста 4</p>	<p>исходное состояние из теста 4;</p> <p>теги во всех подгруппах получают качество GOOD</p>

N	Название теста	Исходное состояние	Требуемое поведение
6	Проверка записи в теги контроллера	OPC-сервер подключен к обоим контроллерам; активен один из контроллеров, в общей подгруппе тегов ROOT.PLC отображаются значения с активного контроллера	при записи в теги общей подгруппы ROOT.PLC или подгруппы ROOT.PLCx активного контроллера, значения меняются синхронно в обеих этих группах; при записи в теги подгруппы ROOT.PLCx неактивного контроллера, значения меняются только в этой группе

В тестах 1-5 все изменения статуса подключения OPC-сервера к контроллерам должны отображаться и адекватно меняться в соответствующих тегах подгруппы ROOT.STATE согласно Руководству по эксплуатации.

Настройки OPC-сервера:

- plc_use_hard_arm_selection = 1.

Таблица 4. Набор тестов

N	Название теста	Исходное состояние	Требуемое поведение
1	Проверка переключения при перезагрузке активного контроллера или обрыве с ним связи	OPC-сервер подключен к обоим контроллерам; активен один из контроллеров, в общей подгруппе тегов ROOT.PLC отображаются значения с активного контроллера	переключения на второй контроллер не происходит, активный контроллер отсутствует; теги в общей подгруппе ROOT.PLC и в подгруппе первого контроллера ROOT.PLCx получают качество BAD; теги в подгруппе второго контроллера ROOT.PLCx сохраняют качество GOOD

N	Название теста	Исходное состояние	Требуемое поведение
2	Проверка переключения при восстановлении связи с контроллером из теста 1	конечное состояние из теста 1	активным снова становится первый контроллер; теги в общей подгруппе ROOT.PLC и в подгруппе первого контроллера ROOT.PLCx получают качество GOOD; в общей подгруппе тегов ROOT.PLC отображаются значения с первого контроллера
3	Проверка переключения через тег ROOT.STATE.set_active_plc	ОПС-сервер подключен к обоим контроллерам; активен один из контроллеров, в общей подгруппе тегов ROOT.PLC отображаются значения с активного контроллера	активным становится второй контроллер; теги в общей подгруппе ROOT.PLC принимают значения со второго контроллера
4	Проверка перезагрузки или обрыва связи с обоими контроллерами	ОПС-сервер подключен к обоим контроллерам; активен один из контроллеров, в общей подгруппе тегов ROOT.PLC отображаются значения с активного контроллера	теги во всех подгруппах получают качество BAD
5	Проверка восстановления связи с обоими контроллерами	конечное состояние из теста 4	исходное состояние из теста 4; теги во всех подгруппах получают качество GOOD
6	Проверка записи в теги контроллера	ОПС-сервер подключен к обоим контроллерам; активен один из контроллеров, в общей подгруппе тегов ROOT.PLC отображаются значения с активного контроллера	при записи в теги общей подгруппы ROOT.PLC или подгруппы ROOT.PLCx активного контроллера, значения меняются синхронно в обеих этих группах; при записи в теги подгруппы ROOT.PLCx неактивного контроллера, значения меняются только в этой группе

В тестах 1-5 все изменения статуса подключения OPC-сервера к контроллерам должны отображаться и адекватно меняться в соответствующих тегах подгруппы ROOT.STATE согласно Руководству по эксплуатации.

Два контроллера с резервированием

Связь между контроллерами и АРМ по одному или двум интерфейсам.

Дерево тегов OPC-сервера:

- ROOT.PLC - теги общей подгруппы активного на данный момент контроллера;
- ROOT.PLCx - подгруппа тегов для работы напрямую с каждым контроллером (x = 1,2);

В OPC-клиенте создаются три OPC-группы с одинаковым набором OPC-тегов из трех подгрупп ROOT.PLC, ROOT.PLC1, ROOT.PLC2.

Таблица 5. Набор тестов

N	Название теста	Исходное состояние	Требуемое поведение
1	Проверка переключения при перезагрузке активного контроллера или обрыве с ним связи	OPC-сервер подключен к обоим контроллерам; активен один из контроллеров, в общей подгруппе тегов ROOT.PLC отображаются значения с активного контроллера	активным становится второй контроллер; теги в общей подгруппе ROOT.PLC принимают значения со второго контроллера; теги в подгруппе первого контроллера ROOT.PLCx получают качество BAD
2	Проверка переключения при восстановлении связи с контроллером из теста 1	конечное состояние из теста 1	активным остается второй контроллер; теги в подгруппе первого контроллера ROOT.PLCx получают качество GOOD
3	Проверка перезагрузки или обрыва связи с обоими контроллерами	OPC-сервер подключен к обоим контроллерам; активен один из контроллеров, в общей подгруппе тегов ROOT.PLC отображаются значения с активного контроллера	теги во всех подгруппах получают качество BAD

N	Название теста	Исходное состояние	Требуемое поведение
4	Проверка восстановления связи с обоими контроллерами	конечное состояние из теста 4	исходное состояние из теста 4; теги во всех подгруппах получают качество GOOD
5	Проверка записи в теги контроллера	<p>ОПС-сервер подключен к обоим контроллерам;</p> <p>активен один из контроллеров, в общей подгруппе тегов ROOT.PLC отображаются значения с активного контроллера</p>	<p>при записи в теги, соответствующие РЕЗЕРВИРУЕМЫМ переменным общей подгруппы ROOT.PLC или подгруппы ROOT.PLCx АКТИВНОГО контроллера, значения меняются синхронно в подгруппе ROOT.PLC и в обеих подгруппах ROOT.PLCx;</p> <p>при записи в теги, соответствующие НЕРЕЗЕРВИРУЕМЫМ переменным общей подгруппы ROOT.PLC или подгруппы ROOT.PLCx АКТИВНОГО контроллера, значения меняются синхронно в обеих этих подгруппах;</p> <p>при записи в теги, соответствующие РЕЗЕРВИРУЕМЫМ переменным подгруппы ROOT.PLCx РЕЗЕРВНОГО контроллера, значения не меняются;</p> <p>при записи в теги, соответствующие НЕРЕЗЕРВИРУЕМЫМ переменным подгруппы ROOT.PLCx РЕЗЕРВНОГО контроллера, значения меняются только в этой подгруппе</p>

N	Название теста	Исходное состояние	Требуемое поведение
6	Проверка синхронного чтения тегов контроллера	<p>ОПС-сервер подключен к обоим контроллерам;</p> <p>активен один из контроллеров, в общей подгруппе тегов ROOT.PLC отображаются значения с активного контроллера</p>	<p>при чтении с указанием в качестве источника контроллера (OPC_DS_DEVICE) производится вычитывание из контроллера;</p> <p>при чтении с указанием в качестве источника кэша данных (OPC_DS_CACHE) неактивного тега производится вычитывание из контроллера;</p> <p>при чтении с указанием в качестве источника кэша данных (OPC_DS_CACHE) тега неактивной группы производится вычитывание из контроллера;</p> <p>при чтении с указанием в качестве источника кэша данных (OPC_DS_CACHE) активного тега активной группы производится отправка значения из кэша</p>

ПРИЛОЖЕНИЕ В. ОГРАНИЧЕНИЕ ПРОПУСКНОЙ СПОСОБНОСТИ

При начальном подключении OPC-сервера к контроллеру и первом вычитывании переменных по каждому из каналов создается высокая нагрузка на процессор, что может привести к потере связи (из-за таймаута ожидания ответа от ПЛК) по остальным каналам подключения (активным или нет) с соседних АРМов и нежелательной многократной смене номера активного канала (и, в конечном счете, контроллера).

Стабильность подключений с нескольких АРМов к одному ПЛК достигается путем увеличения таймаута обмена данными между OPC-сервером и ПЛК (см. п.2.22 Таблица 1). **Важно!** Минусом увеличения значения таймаута обмена является увеличение времени обнаружения обрыва связи по активному каналу – оно составляет **удвоенное значение таймаута обмена**.

Для примера, при фиксированной «нулевой» загрузке процессора ПЛК Regul R500 (пустой проект) и заданном значении таймаута обмена данными 1000 мс получаем следующую таблицу (под клиентом подразумевается один канал связи между OPC-сервером и ПЛК).

Таблица 6. Таблица тестов

Кол-во тегов/кол-во подключений	1000	2000	5000	10000
Время первого цикла опроса при одновременном подключении к ПЛК, сек				
- 1 клиент	~0.7	~0.95	1.6	3.7
- 2 клиента	~0.7	~0.98	1.6/2.0	4.4/5.5
- 3 клиента	~0.7	~0.98	1.8/2.2/2.4	4.3/5.9/7.0
- 5 клиентов	~0.8	~1.0	1.6/2.3/2.6/2.8/3.1	4.6/6.3/7.5/*/* 4.4/6.7/7.9/9.7/12.0 при ТО=4000
Время последующих циклов опроса, сек				
- 1 клиент	~0.2	0.2-0.22	0.61-0.62	1.0-1.2
- 2 клиента	~0.2	0.2-0.22	0.63-0.64	1.0-1.3
- 3 клиента	~0.2	0.21-0.23	0.63-0.67	1.0-1.3
- 5 клиентов	~0.2	0.21-0.23	0.63-0.67	1.0-1.3
Процент загрузки центрального процессора ПЛК				
- 1 клиент	~2%	4-5%	4-5%	3-4%
- 2 клиента	~3%	8-9%	8-9%	7-8%
- 3 клиента	~7%	12-13%	11-13%	11-13%

- 5 клиентов	10-11%	21-26%	18-23%	18-23%
--------------	--------	--------	--------	--------

Как следует из таблицы:

- в наихудшем варианте – при **одновременном** подключении нескольких OPC-серверов к одному контроллеру – при росте числа тегов увеличивается время первоначального считывания и уменьшается стабильность последующих подключений; при 10000 теггах 4-му и 5-му подключению уже не хватает выставленного значения таймаута в 1000 мс; требуется либо увеличить таймаут до 4000 мс, либо подключения установятся при следующих попытках с тем же таймаутом в 1000 мс, когда первые подключения перейдут к циклическому чтению и нагрузка на процессор и среду исполнения CoDeSyS снизится;
- при разнесенных по времени нескольких подключениях OPC-серверов к контроллеру ошибок ожидания таймаута при подключении и первом цикле чтения возникать не должно; но создаваемая при этом нагрузка может привести к таймауту обмена в «соседнем» канале, который уже перешел на циклическое чтение и для него задан небольшой таймаут обмена; в этом случае значение таймаута следует увеличить;
- время циклического чтения не зависит от количества установленных подключений и растет с увеличением числа тегов;
- процент загрузки процессора контроллера не зависит от числа тегов, но растет с увеличением числа подключений.

В рамках работы одного OPC-сервера Regul подключения к одному контроллеру по двум интерфейсам синхронизируются и разнесены по времени. Синхронизации подключений между OPC-серверами Regul, запущенных на разных АРМах нет.

При увеличении стартовой загрузки процессора контроллера проектом все времена вычитывания (и первоначальное и последующие) увеличиваются, соответственно, падает стабильность подключения каналов, что требует соответствующего увеличения значения таймаута обмена.

ПРИЛОЖЕНИЕ Г. ОПИСАНИЕ ФАЙЛА КОНФИГУРАЦИИ

Таблица 7. Параметры конфигурации

№ п/п	Секция/параметр	Описание
1	[options]	описывает общие настройки приложения
1.1	log = 1	включение (1)/ выключение (0) файлового журнала работы программы
1.2	logname = filelog.log	задает базовое имя для лог-файлов и имя для текущего лог-файла. Более старые лог-файлы нумеруются с нарастающей нумерацией от 001 до XXX
1.3	logcount = 10	задает максимальное количество лог-файлов N. Самый старый лог-файл с номером N+1 удаляется
1.4	logsize = 10.0	задает максимальный размер каждого лог-файла в МБ
1.5	confirm_exit = 1	если задано значение 1, то при ручном закрытии приложения будет дополнительно отображаться диалог-запрос подтверждения закрытия
1.6	exit_mode = 1	при заданном значении 0 закрытие приложения будет происходить мгновенно с аварийным закрытием главного процесса. Иначе, выгрузка проводится в штатном режиме с ожиданием всех потоков и освобождением ресурсов
1.7	max_opc_tags_to_send = 5000	задает максимальное количество OPC-тегов, отдаваемых OPC-клиенту по подписке в рамках одной транзакции
1.8	close_on_last_opc_client = 1	если задано значение 1, то приложение будет автоматически выгружено при отключении последнего OPC-клиента (кроме случаев, когда OPC-сервер изначально был запущен вручную)
1.9	opc_client_max_errors = 10	задает максимальное количество ошибок отправки данных OPC-клиенту по подписке (OnDataChange), после которого данная OPC-группа деактивируется в предположении, что OPC-клиент нештатно разорвал подключение
1.10	update_tags_if_only_timestamp_changed = 0	если задано значение 0, то значения тегов будут обновляться при изменении значения (value) или качества (quality) переменной контроллера, изменения только временной метки (timestamp) игнорируются. При установке значения 1 теги будут обновляться даже если изменилась только временная метка (синхронный режим)
1.11	on_startup_hide_in_taskbar = 0	если задано значение 1, то при старте окно приложения будет свернуто в панель управления
1.12	on_startup_hide_in_tray = 0	если задано значение 1, то при старте окно приложения будет свернуто в системный лоток (tray)

№ п/п	Секция/параметр	Описание
1.13	dont_create_params_tree = 0	если задано значение 1, то, после подключения к ПЛК и получения его символьной конфигурации, дерево переменных в окне приложения отображаться не будет
2	[plc]	описывает настройки подключения и режима работы с контроллером
2.1	plc1_port1 = xxx.xxx.xxx.xxx	задают IP-адреса каналов подключения к двум контроллерам (plc1 и plc2) по двум интерфейсам (port1 и port2) для каждого ПЛК; предполагается работа приложения либо с двумя ПЛК в резерве, либо с двумя ПЛК, на которых загружен один и тот же проект; настройка подключения к двум разным ПЛК REGUL или ПЛК с разными проектами НЕДОПУСТИМА
	plc1_port2 = xxx.xxx.xxx.xxx	
	plc2_port1 = xxx.xxx.xxx.xxx	
	plc2_port2 = xxx.xxx.xxx.xxx	
2.2	plc_use_user_login = 0	позволяет подключаться к контроллеру, для которого в Epsilon LD был заведен Online User. Если равен 1, то при подключении используется пара Login/Password из пп.2.3-2.4
2.3	plc_user_login =	имя пользователя (Login) для подключения
2.4	plc_user_password =	пароль пользователя (Password) для подключения
2.5	plc_main = 1	задает номер (1 или 2) ведущего контроллера на момент старта приложения, до момента вычитывания этой информации из самих контроллеров
2.6	plc_main_check_tag = Application.GVL.CPU_Active	задает полное имя переменной в проекте Epsilon LD, определяющей статус ведущего/резервного (True/False) контроллера, в случае наличия связи между контроллерами в схеме с резервированием. Может иметь произвольное название в проекте – оно же задается в этом параметре
2.7	plc_main_check_tag2= Application.GVL.CPU_Default	задает полное имя переменной в проекте Epsilon LD, определяющей статус ведущего/резервного (True/False) контроллера по умолчанию при ошибке синхронизации между двумя контроллерами (в случае, когда у обоих флаг CPU_Active = True). Может иметь произвольное название в проекте – оно же задается в этом параметре
2.8	plc_use_hard_arm_selection = 0	если задано значение 1, то через тег Root.State.set_active_plc происходит безусловный выбор активного контроллера вне зависимости от наличия с ним связи – если связь с ним есть, то данные берутся из него, если связи нет, то все теги в общей группе PLC будут VAD и переключения на другой контроллер не произойдет; значение по умолчанию = 0 – в этом случае OPC-сервер пробует сделать активным контроллер, заданный в Root.State.set_active_plc, если же с ним связи нет, то активным будет второй контроллер (если он на связи), также поведение зависит от параметра 2.9

№ п/п	Секция/параметр	Описание
2.9	plc_set_active_on_link_restore = 0	<p>если значение = 1, то, после восстановления связи с контроллером, заданным в Root.State.set_active_plc, он снова станет активным; по умолчанию значение = 0, при этом:</p> <ul style="list-style-type: none"> – при записи в Root.State.set_active_plc будет выполнен алгоритм как при plc_use_hard_arm_selection = 0 (см. 2.8) – активным будет либо заданный контроллер, либо второй (в зависимости от наличия связи); – если через Root.State.set_active_plc был выбран ПЛК1, затем с ПЛК1 связь пропала, активным стал ПЛК2, затем связь с ПЛК1 восстановилась – активным останется ПЛК2
2.10	plc_enable_app_state_check = 0	<p>если значение = 1, то OPC-сервер дополнительно отслеживает состояние приложения в контроллере (RUN/STOP), которое может быть установлено в режиме отладки из среды Epsilon LD или с помощью переключателя на процессорном модуле. Внимание!!! Рекомендуется использовать только в отладочных целях. На форме настроек в приложении значение инвертировано</p>
2.11	plc1_tags_prefixes= PLC1; TAG_PREFIX_2; TAG_PREFIX_3;...	<p>задает префиксы в именах OPC-тегов, определяющих привязку OPC-группы этих тегов к конкретному контроллеру, в данном случае к первому в конфигурации. Префикс PLC1 задан по умолчанию и соответствует ветке тегов Root.PLC1</p>
2.12	plc2_tags_prefixes= PLC2; TAG_PREFIX_4; TAG_PREFIX_5;...	<p>задает префиксы в именах OPC-тегов, определяющих привязку OPC-группы этих тегов к конкретному контроллеру, в данном случае ко второму в конфигурации. Префикс PLC2 задан по умолчанию и соответствует ветке тегов Root.PLC2</p>
2.13	plc_params_check_period = 500	<p>задает период запроса (частоту обновления) текущих значений всех переменных контроллера (точнее, подмножества переменных, на которые есть подписка от OPC-клиентов) в мс</p>
2.14	plc_reconnect_timeout= 5000	<p>задает паузу в мс после обрыва связи с контроллером перед новой попыткой установления соединения</p>
2.15	plc_link_status_check_period = 100	<p>задает период проверки статуса подключения проводных сетевых интерфейсов сервера (АРМа), на котором запущено приложение OPC-сервера</p>
2.16	plc_browse_enable_arrays = 0	<p>включает/выключает отображение в дереве переменных корневых узлов массивов</p>
2.17	plc_browse_enable_structs = 0	<p>включает/выключает отображение в дереве переменных корневых узлов структур (сложных типов данных)</p>
2.18	plc_logging = 0	<p>включает (1)/ выключает (0) дополнительное логирование работы библиотеки API PLCHandler</p>

№ п/п	Секция/параметр	Описание
2.19	plc_use_server_timestamps = 1	если = 1, то меткой времени значения переменной, считанного из контроллера, является локальное время сервера (АРМа), на котором запущен OPC-сервер (с точностью до миллисекунд). Если = 0, то метка времени предоставляется библиотекой PLCHandler (с точностью до секунд)
2.20	plc_check_alive_counter_tag = Application.GVL.Check_alive_counter	полное имя переменной (здесь Check_alive_counter – как пример) проекта ПЛК, реализующей дополнительный механизм контроля состояния приложения/среды выполнения ПЛК. Переменная должна иметь тип UDINT и представляет собой простой счетчик, инкрементируемый в одной из задач приложения. Переменная должна быть добавлена в символьную конфигурацию проекта. При наличии подключения к контроллеру OPC-сервер периодически вычитывает значение счетчика и, если его значение не меняется в течение периода, заданного в п.2.21, то считается, что приложение на данном контроллере находится в состоянии ошибки (исключения/останова)
2.21	plc_check_alive_counter_to = 500	таймаут проверки изменения переменной счетчика п.2.20, в [мс]
2.22	plc_data_exchange_timeout = 1000	таймаут обмена данными между OPC-сервером и ПЛК в мс; влияет на стабильность каналов подключения к ПЛК на чтение при запуске OPC-сервера на нескольких АРМах
2.23	plc_write_data_exchange_timeout = 1000	таймаут обмена данными между OPC-сервером и ПЛК в мс; для каналов подключения к ПЛК на запись
2.24	plc_opc_lock_timeout = 10	таймаут потока слежения за блокировкой ресурсов в OPC интерфейсе; отладочный параметр
2.25	plc_load_symbols_from_file = 1	включает (1)/выключает (0) сохранение и загрузку символьной конфигурации проекта в файловый кэш; при старте приложения ускоряет загрузку информации о типах и атрибутах доступа переменных символьной конфигурации проекта
2.26	plc_check_symbols_change = 0	включает (1)/выключает (0) проверку изменения символьной конфигурации проекта ПЛК при повторном установлении подключения в канале связи и при переключении на соседний канал связи