

## Программное обеспечение

Резервирование ПЛК NLScon-RSB-XL-CAN-RD в среде Codesys

# Программирование ПЛК NLScon-RSB-XL-CAN-RD с резервированием в среде Codesys

Руководство пользователя

© НИЛ АП, 2024

Версия от 25 сентября 2024 г.

*Одной проблемой стало меньше!*

---

Уважаемый покупатель!

Научно-исследовательская лаборатория автоматизации проектирования (НИЛ АП) благодарит Вас за покупку и просит сообщать нам свои пожелания по улучшению этого руководства или описанной в нем продукции. Ваши пожелания можно направлять по почтовому или электронному адресу, а также сообщать по телефону:

НИЛ АП, пер. Биржевой спуск, 8, Таганрог, 347900,

Тел. (495) 26-66-700,

e-mail: [info@reallab.ru](mailto:info@reallab.ru), <https://www.reallab.ru>.

Вы можете также получить консультации по применению нашей продукции, воспользовавшись указанными выше координатами.

Пожалуйста, внимательно изучите настоящее руководство. Это позволит вам в кратчайший срок и наилучшим образом использовать приобретенное изделие.

Авторские права на программное обеспечение, модуль и настоящее руководство принадлежат НИЛ АП.
--

---

# Оглавление

<b>1. Вводная часть .....</b>	<b>5</b>
1.1. Введение .....	5
1.2. Программные ограничения .....	5
<b>2. Начало работы .....</b>	<b>7</b>
2.1. Установка среды разработки CODESYS V3.5 SP19 Patch 3 .....	7
2.1.1. Минимальные системные требования среды разработки .....	7
2.1.2. Установка компонентов CODESYS .....	8
2.2. Установка таргет-файла RealLab с поддержкой резервирования .....	8
<b>3. Сведения о резервировании ПЛК.....</b>	<b>11</b>
3.1. Схемы соединения ПЛК при резервировании.....	11
3.2. Сетевые настройки ПЛК .....	12
3.3. Общие сведения о резервировании .....	13
3.4. Алгоритм работы резервирования.....	14
3.5. Состояния ПЛК при резервировании.....	18
3.5.1. Штатный режим функционирования резервирования .....	19
3.5.2. Выход из строя ПЛК 1.....	19
3.5.3. Выход из строя ПЛК 2.....	20
3.5.4. Обрыв интерфейса Redundancy Ethernet.....	21
3.5.5. Состояние резервирования: Ошибка запуска.....	21
3.5.6. Состояние резервирования: Запуск.....	21
3.5.7. Состояние резервирования: Автономный на ПЛК 1 и ПЛК 2. .	22
3.5.8. Потеря связи между средой CoDeSys и ПЛК .....	22
3.6. Индикация ПЛК с резервированием .....	23
3.6.1. Индикация состояния исполнительной среды CODESYS .....	23
3.6.2. Индикация состояния резервирования .....	23
<b>4. Резервирование ПЛК в среде Codesys .....</b>	<b>23</b>

---

4.1. Настройка резервирования в Codesys.....	23
4.1.1. Создание проекта в Codesys с помощью шаблона .....	23
4.1.2. Создание проекта в Codesys с нуля.....	26
4.1.3. Настройка параметров резервирования.....	27
4.1.4. Запуск ПЛК .....	30
4.1.5. Функция авто-синхронизации данных между ПЛК.....	32
4.2. Заводской сброс ПЛК в Codesys.....	33
4.3. Работа переключателя Старт/Стоп.....	35
<b>5. Реализация дополнительных функций в CODESYS ....</b>	<b>35</b>
5.1. Веб-визуализация .....	35
5.2. Работа светодиодов состояния резервирования ПЛК .....	38
5.3. Работа с Retain-переменными .....	39
5.4. Управление дискретными входами/выходами (GPIO).....	40
<b>Лист регистрации изменений.....</b>	<b>43</b>

# 1. Вводная часть

## 1.1. Введение

Основной функцией резервирования является исполнение одного и того же приложения стандарта МЭК 61131-3 на двух независимых (ПЛК) одновременно. При отказе основного ПЛК активируется резервный и безударно перехватывает управление. Функция обеспечивает взаимный мониторинг и синхронизацию двух контроллеров.

Избыточное перекрестное соединение между двумя контроллерами реализовано с использованием протокола на основе TCP/UDP. Активный контроллер управляет системой ввода-вывода, в то время как пассивный контроллер контролирует активный контроллер и синхронизируется с ним. При запуске избыточной системы первый запущенный ПЛК переходит в состояние «автономный» до тех пор, пока не будет запущен и синхронизирован второй ПЛК. Затем первый переходит в активное состояние, второй — в пассивное (резервное). В случае сбоя управления происходит автоматическое переключение.

Ручное переключение может быть инициировано через функциональный интерфейс. Загрузочное приложение и память обновляются автоматически. Таймеры IEC синхронизируются в каждом цикле задачи.

## 1.2. Программные ограничения

**CODESYS Redundancy:** для корректной работы требуется среда разработки CODESYS версии не ниже *3.5.17.10*. На обоих устройствах ПЛК должна быть установлена одна и та же версия системы исполнения.

**Синхронизация задач в реальном времени:** поддерживается синхронизация не более 1 задачи в реальном времени. Синхронизация большего количества задач возможна, но все остальные задачи будут выполняться без синхронизации на обоих ПЛК (кроме одной, которая синхронизируется в режиме реального времени).

**Выполнение задачи в реальном времени** означает, что задача резервирования имеет ограниченную задержку. Связь в реальном времени означает, что сообщение, отправленное посредством резервного соединения, принимается вторым ПЛК в течение определенного времени.

**Тайм-аут** резервирования можно определить в системе разработки CODESYS с помощью редактора конфигурации резервирования (в параметрах резервирования).

При необходимости резервирования нескольких программ, используйте резервирование только для основной программы, а работу с остальными задачами производите с помощью списка глобальных переменных.

**МЭК-таймер:** разное время выполнения задачи на обоих ПЛК может вызывать отклонения выходных значений при переключении ПЛК. Чтобы предотвратить это, значения таймера МЭК замораживаются во время выполнения задачи. Таким образом, вызовы таймеров МЭК (пример: TON) при выполнении задачи всегда приводят к одним и тем же значениям таймера, даже если физическое время продолжает идти. Это означает, что невозможно, например, использовать активное ожидание в цикле, поскольку значения таймера МЭК не изменяются при текущем сканировании задачи.

**Указатель:** переменные-указатели не должны использоваться в резервируемой области данных. Значения, управляемые резервированием, передаются в другой ПЛК во время синхронизации. Однако значения указателей недопустимы на другом ПЛК, поскольку там может быть другая структура памяти.

В случае использования указателя в резервной области среда разработки показывает предупреждение для каждой используемой переменной-указателя.

Отключить данную проверку можно с помощью добавление следующей записи в xml-файл(таргет файл) описания устройства:

```
<Device>  
<Custom>  
<Redundancy DisablePointerChecks="true">
```

**Map on existing:** метод сопоставления ввода-вывода «Map on Existing» (сопоставление ввода-вывода с существующими переменными) не рекомендуется использовать с CODESYS Redundancy. Переменные такого типа хранятся не в областях входных или выходных данных, а там, где они объявлены. В результате они не синхронизируются во время работы.

Сетевые переменные с доступом для записи нельзя использовать, поскольку одновременно отправляется несколько телеграмм записи. Сетевые переменные с доступом для чтения разрешены.

## Начало работы

---

**Доступ к файлам:** доступ к файлам использовать не рекомендуется, поскольку разные данные внутри файлов могут вызвать проблемы на контроллерах при их переключении.

Если вы используете файлы, вам необходимо объявить дескрипторы файлов в областях данных, которые не контролируются резервированием. Файлы необходимо открывать отдельно на обоих ПЛК. Дескриптор файла другого ПЛК нельзя использовать для доступа к файлам на локальном ПК.

Во время компиляции проверяются идентификаторы переменных (RTS\_IEC\_HANDLE, CAA.HANDLE), расположенные в области, контролируемой резервированием. Предупреждение выдается для каждой переменной-дескриптора, обнаруженной в такой области.

**Управление безопасностью подключения пользователей:** если используется онлайн-управление безопасностью подключения пользователей ПЛК, вам необходимо настроить оба ПЛК с одинаковыми именами пользователей и паролями. В противном случае онлайн-сервисы, такие как “запись переменных” или “онлайн-замена” не будут взаимодействовать с пассивным ПЛК.

**SoftMotion:** CODESYS SoftMotion и CODESYS Redundancy нельзя комбинировать. Требования времени SoftMotion не могут быть выполнены при использовании резервирования.

## 2. Начало работы

### 2.1. Установка среды разработки CODESYS V3.5 SP19 Patch 3

#### 2.1.1. Минимальные системные требования среды разработки

Для корректной работы среды разработки CoDeSys IDE требуется:

- операционная система Windows 10 или выше;
- 8 Гб оперативной памяти;
- 12 Гб свободного места на HDD;
- процессор: 2,5 GHz.

Для того, чтобы установить CODESYS 3.5.19.30, скачайте программу установки с сайта и запустите. Далее следуйте указаниям установщика рис. 2.1. В процессе установки есть возможность выбора основных компонентов для установки.

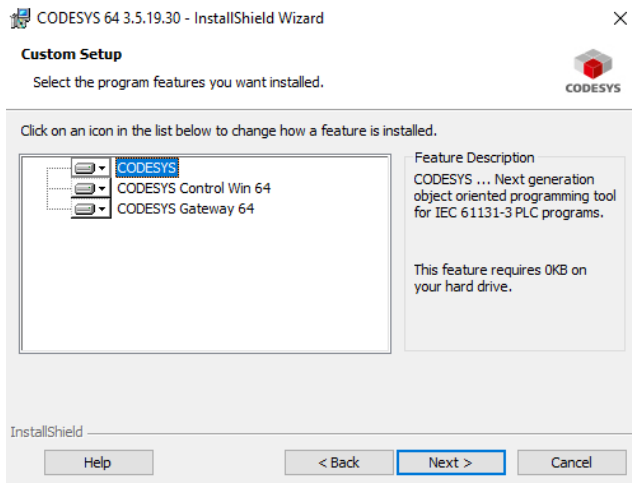


Рис. 2.1. Окно установки CODESYS

### 2.1.2. Установка компонентов CODESYS

Существуют компоненты, которые устанавливаются отдельно, установка производится с помощью среды разработки CODESYS. Скачайте с нашего [сайта](#) необходимые пакеты установки и установите их следующим образом.

Запустите среду разработки Codesys 3.5 и из выпадающего меню “Инструменты” выберите “CODESYS Installer...”. Откроется окно со списком компонентов, установленных в CODESYS этой версии. Нажмите Install и выберите ваш пакет в формате “.package” для установки.

Закройте все окна среды разработки CODESYS, кроме CODESYS Installer. Следуйте инструкциям установщика. По окончании процесса откройте CODESYS и проверьте работоспособность установленных пакетов. Более подробно процесс установки описан в п. 2.2.

## 2.2. Установка таргет-файла RealLab с поддержкой резервирования

Для того чтобы установить таргет-файл для ПЛК с поддержкой резервирования необходимо загрузить его с нашего [сайта](#). Если у Вас не установлены пакеты “CODESYS Edge Gateway for Linux” и “CODESYS Control for Linux



## Начало работы

---

SL”, их также нужно установить – они поставляются в архиве вместе с пакетом, содержащим таргет-файл. Распакуйте скачанный архив в любое место на накопителе компьютера.

Запустите среду разработки CODESYS. В верхнем меню (рис. 2.2) выберите “Инструменты – CODESYS Installer”.

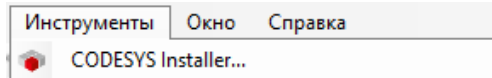


Рис. 2.2. Запуск CODESYS Installer

Откроется новое окно (рис. 2.3). Для установки новых пакетов нажмите “Install File”.

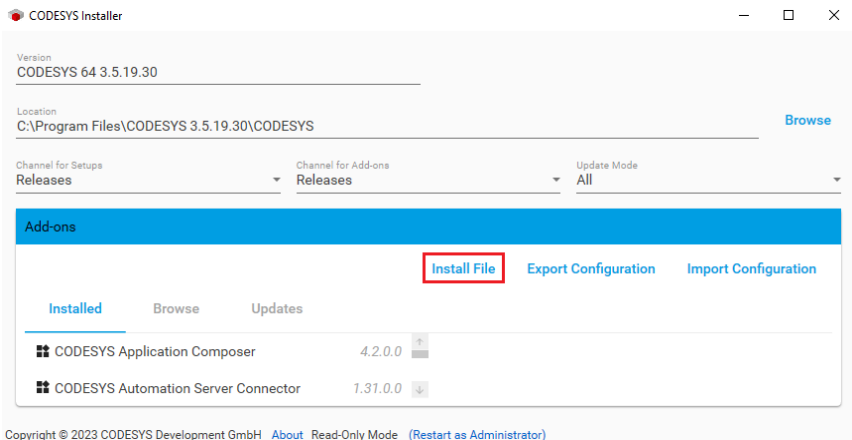


Рис. 2.3. Окно CODESYS Installer

Закройте окно среды разработки CODESYS. Нажмите на кнопку “Install File” и выберите пакет для установки. В процессе установки откроется новое окно (рис. 2.4), в нём нужно отметить все галочки и нажать “Continue”.

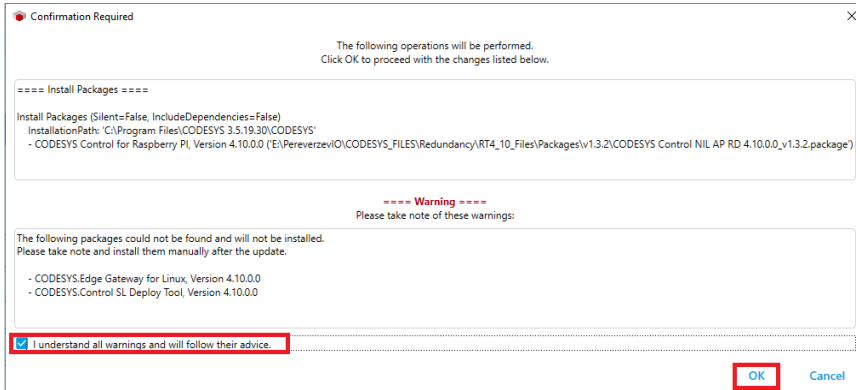


Рис. 2.4. Установка галочки

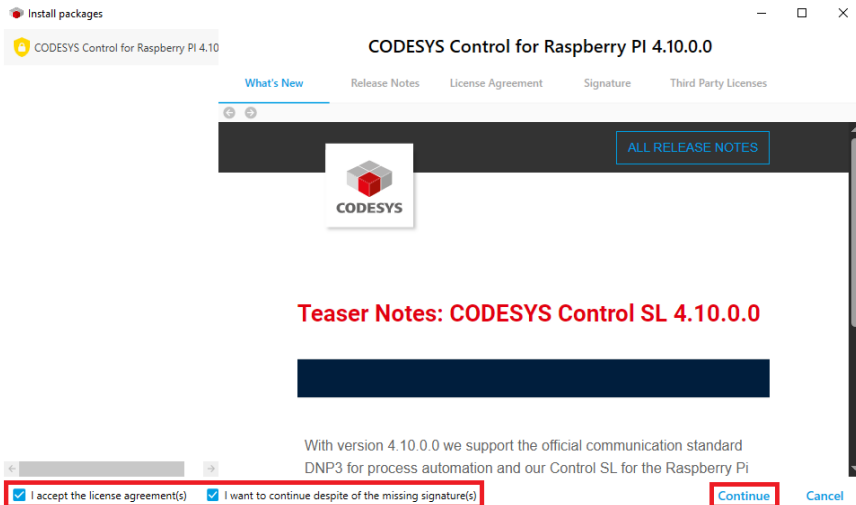


Рис. 2.5. Продолжение установки

После установки появится сообщение о том, что пакет успешно установлен (рис. 2.6). Повторите данные действия для всех пакетов (.package), которые нужно установить.

The add-ons were successfully installed.

Рис. 2.6. Успешное завершение установки

## 3. Сведения о резервировании ПЛК

### 3.1. Схемы соединения ПЛК при резервировании

ПЛК (ПЛК N) позволяет устанавливать соединение с модулями по протоколу CAN, Modbus TCP. Ethernet устройства можно подключить к коммутатору или к свободному Ethernet-порту ПЛК (при условии что порт настроен на нужную подсеть). CAN и Modbus-модули подключаются к портам ПЛК.

Интерфейс eth0 по умолчанию имеет статический IP-адрес и используется для связи и синхронизации данных между основным и резервным ПЛК.

Интерфейс eth1 по умолчанию имеет динамический IP-адрес и используется для связи со средой разработки или системой верхнего уровня.

Настройка IP-адресов интерфейсов описана в п. 3.2.

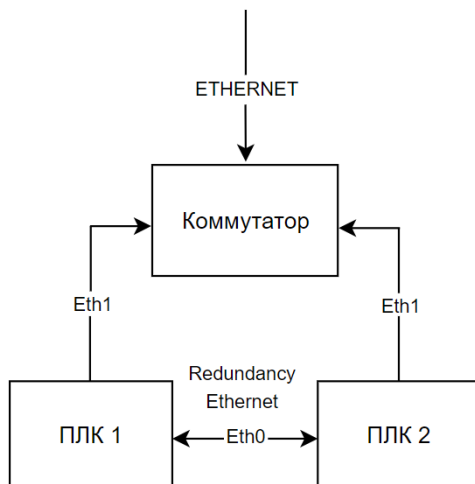


Рис. 3.1. Схема 1

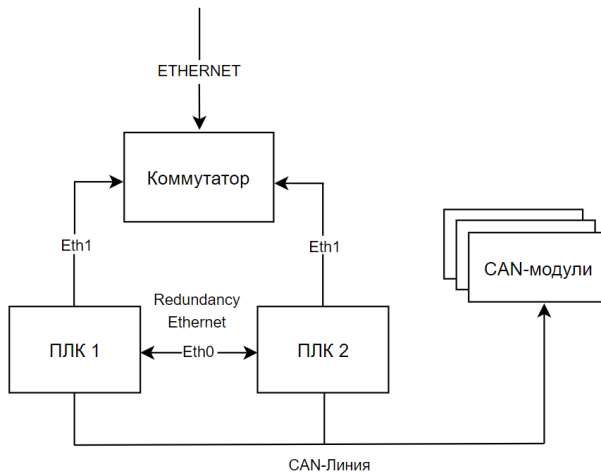


Рис. 3.2. Схема 2

### 3.2. Сетевые настройки ПЛК

Подключитесь к ПЛК через VNC или SSH. Для настройки IP-адресов интерфейсов eth0 и eth1 воспользуйтесь встроенным в ПЛК консольным конфигуратором – “rl-config”.

Запустите конфигуратор терминальной командой “sudo rl-config”:

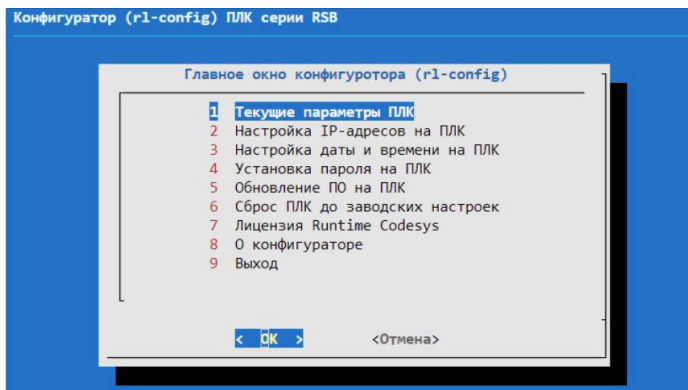


Рис. 3.3. Конфигуратор rl-config

## Сведения о резервировании ПЛК

---

Выберите пункт 2 – “Настройка IP-адресов на ПЛК”.

Из списка выберите интерфейс для настройки (eth0 или eth1).

Введите значения для IP-адреса, маски, DNS-серверов.

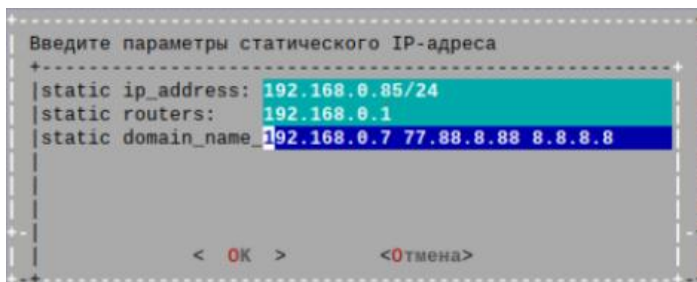


Рис. 3.4. Пример настройка статического IP-адреса

Сохраните результат изменения, нажав на «Ок» и подтвердив данные изменения. Проверить результат изменения IP адреса можно в конфигураторе в разделе “Текущие параметры ПЛК”.

Более подробная настройка статического IP-адреса описана в руководстве эксплуатации ПЛК.

### 3.3. Общие сведения о резервировании

Горячее резервирование подразумевает наличие двух ПЛК. Первый ПЛК (Активный или основной) работает с модулями, а второй синхронизирует данные с первым ПЛК и находится в резервном (Пассивном) состоянии. Если первый ПЛК выйдет из строя (отключится, или связь с ним прервётся), второй ПЛК возьмет на себя роль основного, но из-за отсутствия ПЛК, с которым возможно синхронизироваться, второй ПЛК будет находиться в состоянии “Автономный”.

ПЛК могут находиться в одном из следующих состояний:

- **Инициализация** – состояние ПЛК, в котором он находится в момент запуска, перезагрузки, или после восстановления связи.
- **Синхронизация** – состояние ПЛК, в котором он находится во время синхронизации с другим ПЛК.

- **Активный** – штатное состояние основного ПЛК в резервной паре. В этом состоянии ПЛК выполняет пользовательское приложение, осуществляет управление технологическим процессом и отправляет сообщения второму (пассивному) ПЛК данные для синхронизации.
- **Пассивный** – штатное состояние резервного ПЛК в резервной паре. ПЛК получает загрузочное приложение от Активного, загружает его и синхронизирует выполнения задачи. Пассивный в любой момент готов заменить активный ПЛК.
- **Автономный** – нештатное состояние ПЛК. Один из контроллеров переходит в это состояние, если другой ПЛК вышел из строя или связь с ним разорвана, то есть нет ПЛК, с которым можно синхронизировать выполнение задачи.
- **Эмуляция** – нештатное состояние ПЛК. Обусловлено отсутствием обмена между ПЛК или ошибками на шине CAN. Также можно перевести ПЛК в это состояние намеренно.
- **Ошибка синхронизации** – возникает при сбое в момент синхронизации ПЛК. Например, при несоответствии областей синхронизируемых данных или при наличии в проекте более одной задачи для синхронизации.
- **Запуск** – состояние ПЛК, в котором он находится до инициализации. Если ПЛК долго находится в этом состоянии, оно приравнивается к Ошибке Синхронизации.
- **Ошибка цикла** – нештатное состояние ПЛК. Проявляется при возникновении исключения на одном из ПЛК.

Подробнее о причинах изменения состояний ПЛК в таблице табл. 1.

### 3.4. Алгоритм работы резервирования

#### Инициализация

Состояние обоих ПЛК определяется на этапе запуска. Если ПЛК запуск осуществляется по очереди, то первый запущенный будет Активным, второй – Пассивным. Если ПЛК запущены одновременно, то их роли распределяются в зависимости от их ID:

- ID 1 – Основной ПЛК – Активный.
- ID 2 – Резервный ПЛК – Пассивный.

ID присваиваются и записываются в ПЛК при добавлении их в среду разработки (см. рис. 4.6).

## Сведения о резервировании ПЛК

---

### Этап 1. Загрузка.

В первую очередь проект загружается в Основной ПЛК (Активный). Затем он отправляет сообщение о загрузке Резервному (Пассивному) ПЛК. После отправки Основной ПЛК принимает роль Автономного, а второй ПЛК начинает загружаться.

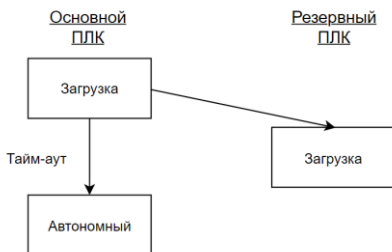


Рис. 3.5. Диаграмма загрузки ПЛК

### Этап 2. Синхронизация.

Как только Основной ПЛК получит сообщение о завершении загрузки, он отправит ответное сообщение о видимости и загрузочный файл. Резервный ПЛК начинает процесс синхронизации.

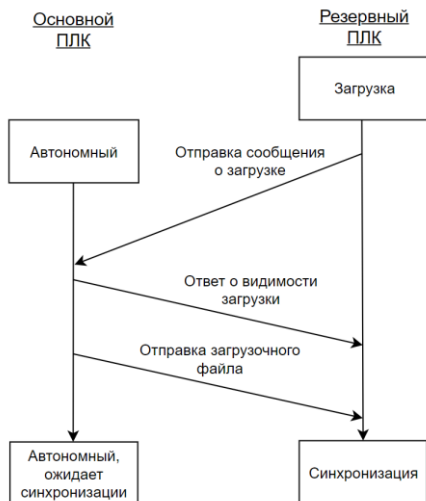


Рис. 3.6. Диаграмма начала синхронизации ПЛК

### Этап 3. Завершение синхронизации.

Основной ПЛК, ожидая завершения синхронизации, начинает выполнение первого цикла программы. Завершив синхронизацию, Резервный ПЛК отправляет об этом сообщение и также начинает выполнение пользовательской программы. Получив сообщение, Основной ПЛК отправляет ответ о видимости синхронизации. Завершив синхронизацию, Основной ПЛК переключается в Активное состояние, а Резервный ПЛК – в Пассивное.

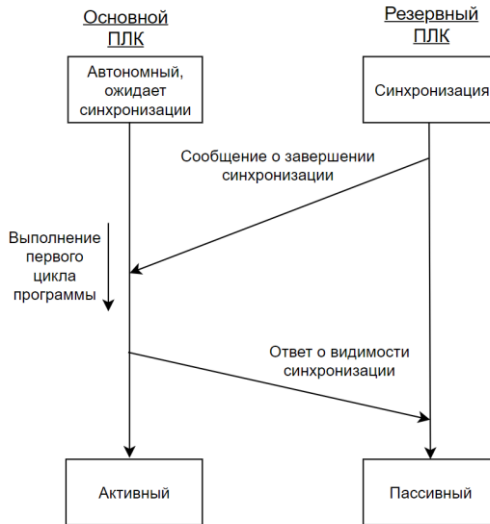


Рис. 3.7. Завершение синхронизации

Теперь резервирование активно и задача приложения синхронно выполняется на обоих ПЛК.

### Штатный режим работы резервирования

Время начала очередного цикла задачи резервирования различается для Активного и Пассивного ПЛК из-за погрешности планировщика задач, разница значений может составлять несколько миллисекунд. Контроллеры обмениваются сообщениями о начале и завершении выполнения цикла программы.



## Сведения о резервировании ПЛК

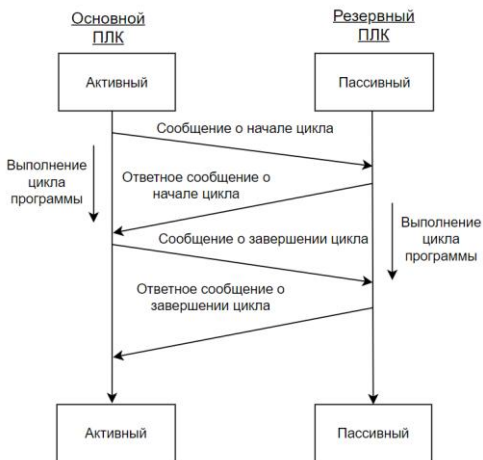


Рис. 3.8. Штатный режим работы резервирования

Если связь **Redundancy Ethernet** между ПЛК разорвана, то Основной и Резервный ПЛК переходят в Автономные состояния.

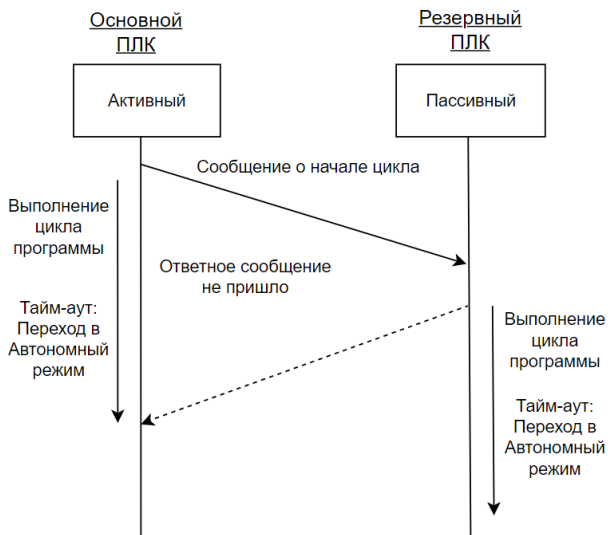


Рис. 3.9. Переход в Автономный режим

### 3.5. Состояния ПЛК при резервировании

Варианты состояний контроллеров при резервировании и причины их изменения приведены в табл. 1.

Табл. 1. Состояния контроллеров

Исходное состояние		После события		Причина
ПЛК 1	ПЛК 2	ПЛК 1	ПЛК 2	
-	-	Активный	Пассивный	Штатный режим работы
-	-	Пассивный	Активный	
Активный	Пассивный	Пассивный	Активный	Сбой в активном ПЛК
Пассивный	Активный	Активный	Пассивный	
Активный	Пассивный	-	Автономный	Выход из строя ПЛК 1
Пассивный	Активный	-	Автономный	
Активный	Пассивный	Автономный	-	Выход из строя ПЛК 2
Пассивный	Активный	Автономный	-	
Активный	Пассивный	Автономный	Автономный	Обрыв по интерфейсу Redundancy Ethernet
Пассивный	Активный	Автономный	Автономный	
Активный	Пассивный	Ошибка цикла	Автономный	Исключение в программе на одном из ПЛК
Пассивный	Активный	Автономный	Ошибка цикла	
-	-	Автономный	Ошибка запуска	Не загружено приложение пользователя с резервированием в ПЛК 2
-	-	Автономный	Запуск	Неверно указаны IP-адреса для резервирования
-	-	Автономный	Автономный	Неверно настроено резервирование или среда CODESYS

### 3.5.1. Штатный режим функционирования резервирования

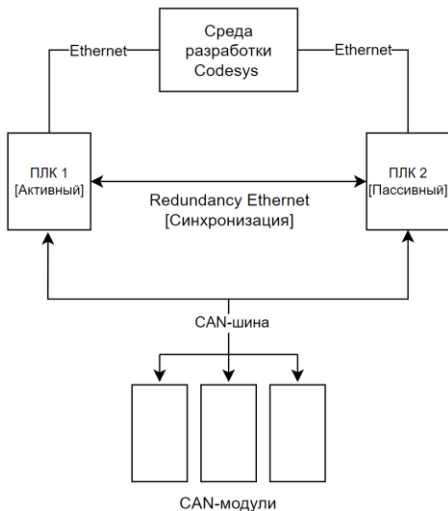


Рис. 3.10. Штатный режим работы резервирования ПЛК

### 3.5.2. Выход из строя ПЛК 1

В случае выхода из строя ПЛК 1 управление технологическим процессом переходит к ПЛК 2, который меняет своё состояние на Автономный.

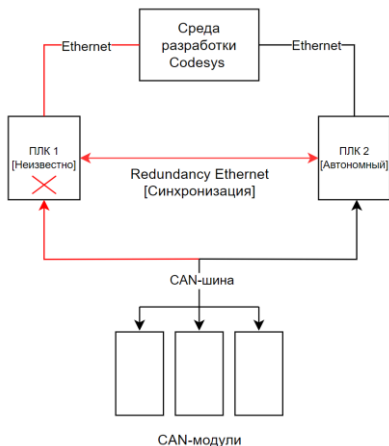


Рис. 3.11. Выход из строя ПЛК 1

После отказа Активного ПЛК и последующего возвращения в работу он станет Пассивным.

### 3.5.3. Выход из строя ПЛК 2

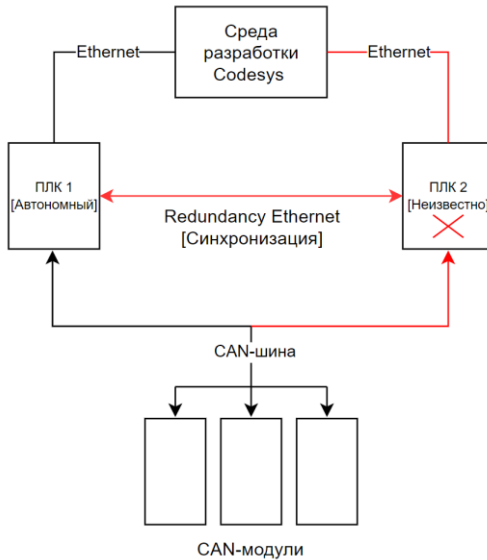


Рис. 3.12. Выход из строя ПЛК 2

В случае выхода из строя ПЛК 2, в среде CoDeSys ему присваивается состояние «Неизвестно», а ПЛК 1 продолжит управлять технологическим процессом и станет Автономным. Пользователь сможет осуществлять мониторинг выполнения программы в среде разработки CoDeSys.

### 3.5.4. Обрыв интерфейса Redundancy Ethernet

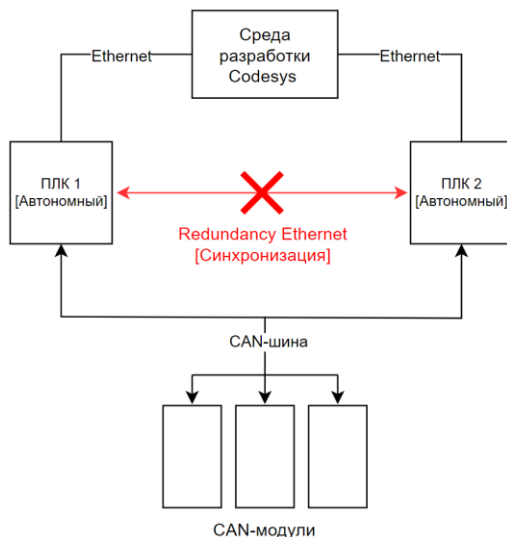


Рис. 3.13. Обрыв интерфейса Redundancy Ethernet

В случае обрыва резервирования по интерфейсу Redundancy Ethernet оба ПЛК перейдут в Автономный режим. После восстановления соединения ПЛК 1 перейдет в Активное состояние, а ПЛК 2 – в Пассивное.

### 3.5.5. Состояние резервирования: Ошибка запуска

При первом запуске, если ранее в контроллеры не были загружены пользовательские приложения с поддержкой резервирования, ПЛК 2 не сможет синхронизироваться автоматически и перейдет в состояние резервирования «Ошибка запуска». Нужно нажать на кнопку «Синхронизация» в поле ПЛК 2 и «Состояние резервирования» должно смениться на «Активно» и «Пассивный» ПЛК 1 и ПЛК 2 соответственно.

### 3.5.6. Состояние резервирования: Запуск

Данное состояние является штатным, но только на этапах Инициализации. Если ПЛК находится в этом состоянии продолжительное время, то в данном случае имеют место неправильные настройки IP-адресов ПЛК 1 и ПЛК 2. Для решения проблемы нужно проверить правильность настроек компонента «Конфигурация резервирования».

### 3.5.7. Состояние резервирования: Автономный на ПЛК 1 и ПЛК 2.

Причиной подобного состояния могут являться следующие неправильные настройки:

1. Активные пути для ПЛК 1 и ПЛК 2 на вкладке «Состояние резервирования» были установлены на один и тот же CPU,
2. Выбраны одинаковые роли для ПЛК 1 и ПЛК 2.

Для решения проблемы необходимо скорректировать настройки компонента “Конфигурация резервирования”.

### 3.5.8. Потеря связи между средой CoDeSys и ПЛК

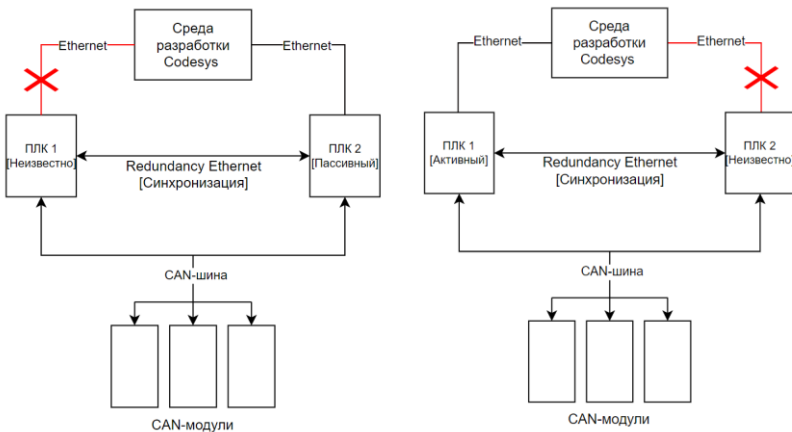


Рис. 3.14. Потеря связи между Codesys и ПЛК

Потеря связи между средой CoDeSys и ПЛК никак не влияют на состояния резервирования ПЛК 1 и ПЛК 2.

При потере связи с ПЛК 2(1) из штатного режима работы резервирования, его состояние в среде Codesys изменится на “Неизвестно”, но фактическое будет Пассивным (Активным). Если произойдёт нештатная ситуация, и ПЛК 1(2) выйдет из строя, то ПЛК 2(1) возьмёт на себя управление технологическим процессом.

### 3.6. Индикация ПЛК с резервированием

#### 3.6.1. Индикация состояния исполнительной среды CODESYS

Индикатор на корпусе ПЛК: Старт/Стоп.

- Переключатель Run-Stop, останавливает/запускает МЭК программу MasterSCADA 4D;
- При запуске МЭК программы светодиод горит зеленым цветом, а при остановке – красным.
- При запуске ПЛК МЭК-программа проверяет состояние переключателя. Если переключатель в положении Стоп, то МЭК-программа останавливается, а если в переключатель в положении Старт, то запускается.

#### 3.6.2. Индикация состояния резервирования

Табл. 2. Индикация состояний резервирования

Состояние резервирования	Цвет и состояние индикатора
Активный	Зеленый (постоянное свечение)
Пассивный	Красный (постоянное свечение)
Автономный	Зеленый (мигание с интервалом 250 мс)
Эмуляция	Красный (мигание с интервалом 1000 мс)
Ошибка синхронизации	Красный (мигание с интервалом 250 мс)

## 4. Резервирование ПЛК в среде Codesys

### 4.1. Настройка резервирования в Codesys

#### 4.1.1. Создание проекта в Codesys с помощью шаблона

Чтобы быстро создать проект, в котором реализована индикация состояний резервирования, а также функционал автоматического выхода ПЛК из состояния эмуляции и т.д., достаточно при создании нового проекта выбрать шаблон “Redundancy RealLab Project”. Шаблон также позволяет управлять индикацией состояний резервирования.

Откройте среду разработки Codesys. На стартовой странице выберите “Новый проект...”.

Выберите шаблон “Redundancy RealLab Project”. Укажите имя проекта и его расположение. Нажмите “ОК”.

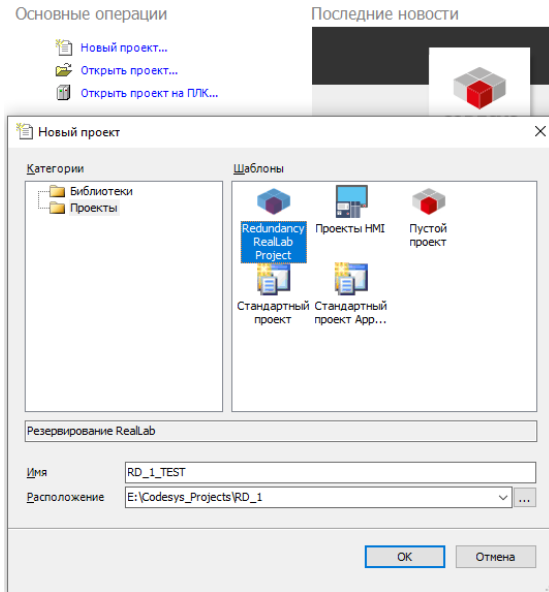


Рис. 4.1. Выбор шаблона резервирования для создания проекта

Далее созданный на основе шаблона проект откроется в Codesys. Среда разработки предложит удалить внутренние данные об удаленных retain-переменных – нажмите “Да”.

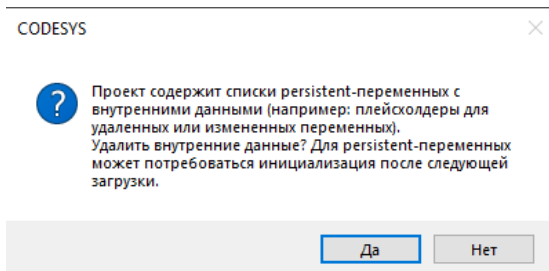


Рис. 4.2. Вопрос CODESYS о Retain-переменных



## Резервирование ПЛК в среде Codesys

Дерево проекта (см.рис. 4.3) содержит:

1. Устройство (ПЛК);
2. Два файла программ на ST (управление светодиодами состояния в файле LEDs, файл с основной резервируемой программой POU содержащий код синхронизации и получения диагностических данных).
3. Конфигурация резервирования (требуется настроить IP-адреса ПЛК);
4. Конфигурация задач;
5. Список retain-переменных;
6. CANbus и CANopenManager;
7. GPIO для управления дискретными входами и выходами, а также светодиодами состояния резервирования ПЛК.

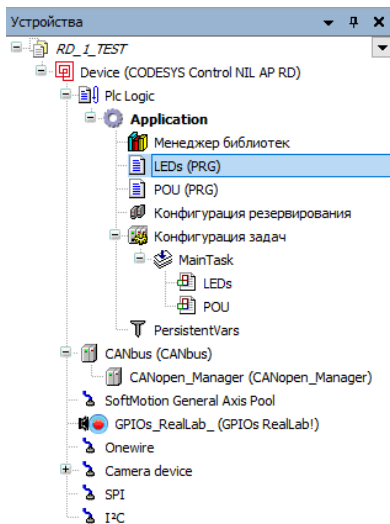


Рис. 4.3. Дерево шаблонного проекта

Для того, чтобы установить параметры резервирования и выполнить загрузку проекта, используйте п. 4.1.3 и п. 4.1.4. Для настройки retain-переменных используйте п. 5.3.

**Важно!** При работе с CAN-модулями и протоколом CANopen отключите параметр “Генерация сердцебиения” в параметрах CANopen\_Manager, иначе функционал резервирования будет нарушен.

### 4.1.2. Создание проекта в Codesys с нуля

Запустите CODESYS. Создайте стандартный проект с устройством “CODESYS Control NIL AP RD” (см. рис. 4.4).

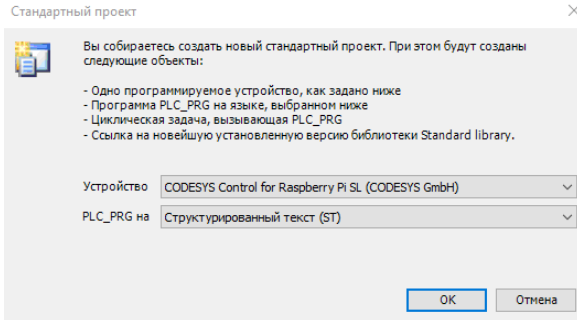


Рис. 4.4. Выбор устройства при создании проекта.

Далее кликните правой кнопкой мыши по вкладке “Application” в дереве устройств. Откроется контекстное меню. Выберите добавление объекта – конфигурация резервирования (см. рис. 4.5). В диалоговом окне нажмите “Добавить”.

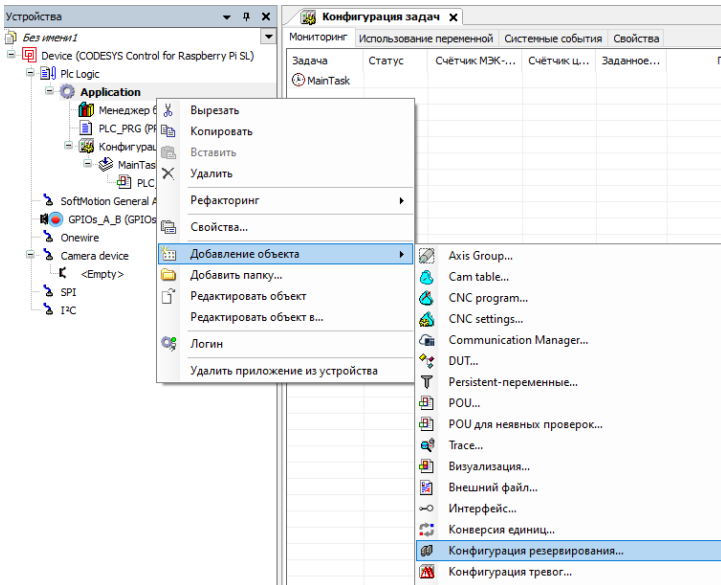


Рис. 4.5. Добавление конфигурации резервирования

### 4.1.3. Настройка параметров резервирования

Дважды кликните по появившейся “Конфигурации резервирования” в дереве устройств. Откроется окно с активной вкладкой “Состояние резервирования”. Нажмите “Путь ПЛК 1...”, Вас перебросит в окно “Установки соединения”. Нажмите “Сканировать сеть” и выберите основной ПЛК, ему автоматически присвоится ID=0. Затем вернитесь к конфигурации резервирования и нажмите “Путь ПЛК 2” и выберите резервный ПЛК, ему автоматически присвоится ID=1.

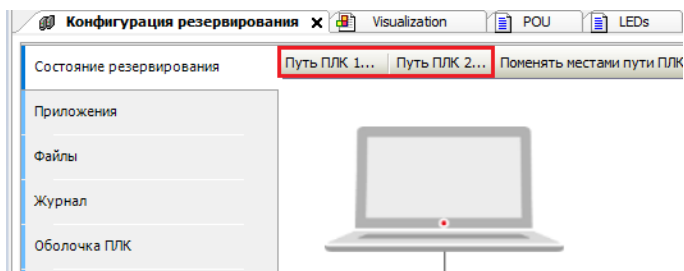


Рис. 4.6. Установка пути ПЛК

Убедитесь, что контроллеры соединены друг с другом и подключены к сети Ethernet в соответствии со схемой в п. 3.1. Затем перейдите во вкладку “Параметры Резервирования”.

Для резервной связи установите IP-адреса интерфейсов eth0 обоих ПЛК.

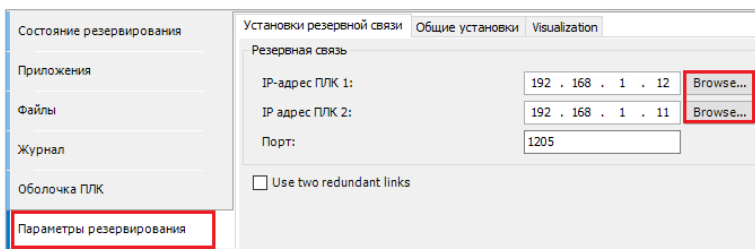


Рис. 4.7. Настройка резервной связи между ПЛК

Во внутренней вкладке “общие установки” можно настроить таймаут синхронизации и прочие параметры.

**Таймаут (мс)** – это максимальное время, в течение которого ожидается ответ на запрос от активного контроллера. Если в течение этого времени ответ не получен, то происходит переключение ролей. Оптимальное значение таймаута должно быть больше или равно времени цикла резервной задачи.

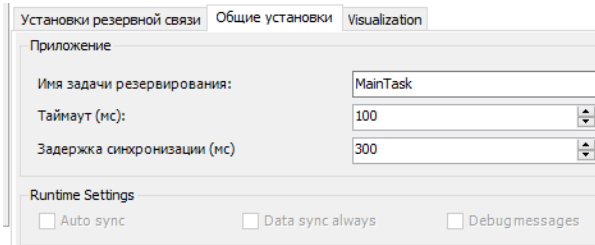


Рис. 4.8. Общие установки резервирования

Например, если  $\text{MainTask} = 100$  мс, то таймаут переключения - минимум 100 мс.

Таймаут синхронизации / Задержка синхронизации (мс) – время ожидания резервного контроллера для синхронизации с активным. Минимальное значение должно быть равно таймауту переключения.

Для того чтобы резервировать визуализацию, перейдите во внутреннюю вкладку Visualization. Поставьте галочку напротив “Enable”. Настройте IP-адреса для резервирования визуализации и установите остальные параметры по необходимости.

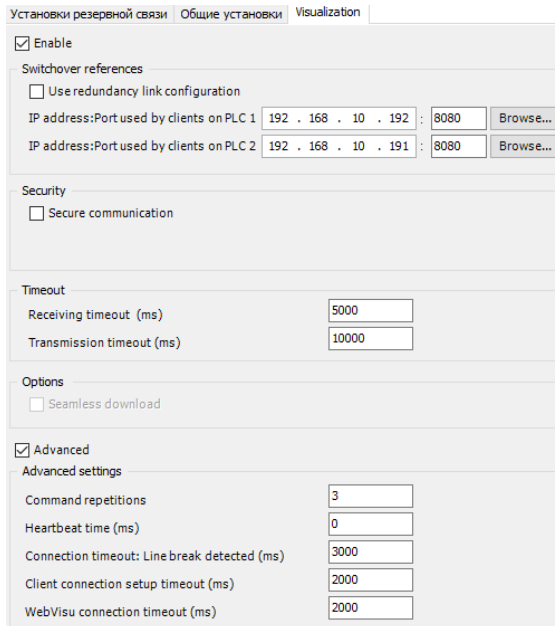


Рис. 4.9. Параметры резервирования визуализации

## Резервирование ПЛК в среде Codesys

---

После настройки нажмите “Запись” в нижней части окна – так вы запишите настройки в ваши резервируемые ПЛК. После применения настроек на обоих ПЛК нужно перезагрузить среду исполнения.

**Важно!** Перед перезагрузкой исполнительной среды на ПЛК проверьте положение переключателя “Старт/Стоп” на обоих ПЛК – он должен быть в положении “Старт”!

Перезагрузку необходимо произвести через среду разработки CODESYS. В CODESYS в верхнем меню выберите “Инструменты – Update Raspberry Pi”.

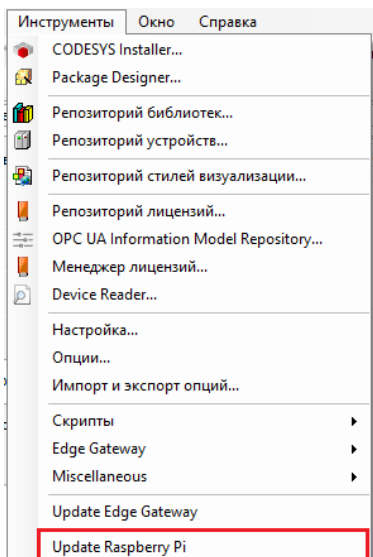


Рис. 4.10. Выбор пункта из инструментов CODESYS

В левой части экрана откроется меню для управления ПЛК.

Процесс перезагрузки среды исполнения:

- Введите имя пользователя (pi), пароль (123) и IP-адрес ПЛК 2 (рис. 4.11);
- Нажмите “Stop” для остановки исполнительной среды на ПЛК 2. То же самое повторите с ПЛК 1;
- Запустите исполнительную среду на ПЛК 1 нажатием на “Start” в меню управления ПЛК;
- Выполните логин с загрузкой проекта;
- Запустите ПЛК 2 нажатием “Start” в меню управления ПЛК.

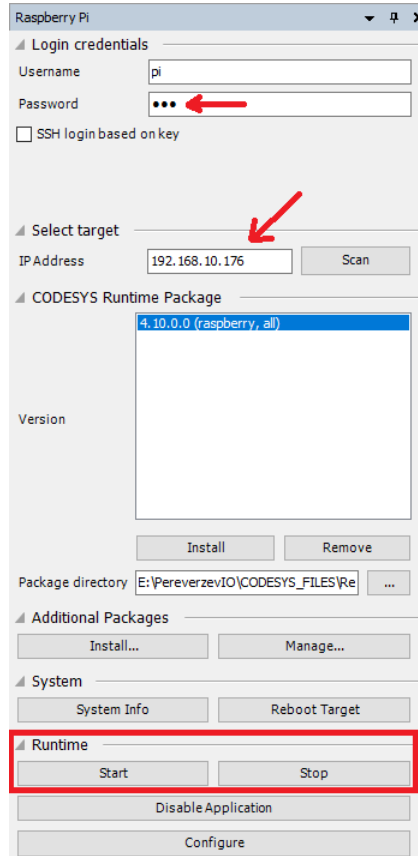


Рис. 4.11. Меню управления ПЛК

### 4.1.4. Запуск ПЛК

После перезагрузки ПЛК, подключитесь к нему и выполните загрузку проекта. После этого нажмите на значок, который позволит отслеживать состояния обоих ПЛК.

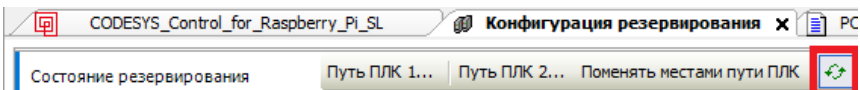


Рис. 4.12. Кнопка обновления состояния ПЛК

Один из контроллеров станет активным, а другой – пассивным.

## Резервирование ПЛК в среде Codesys

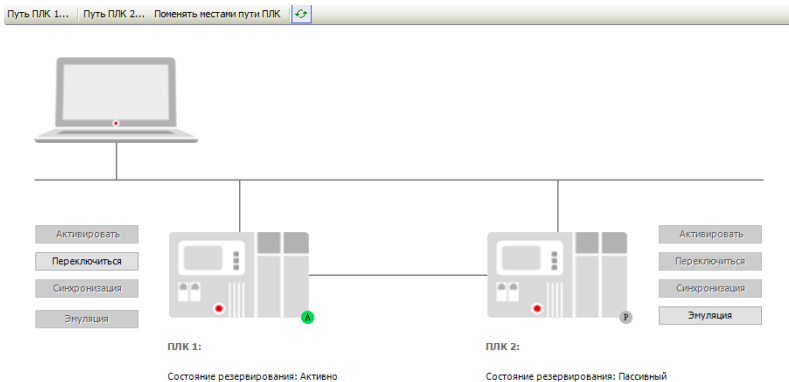


Рис. 4.13. Состояния ПЛК

В “Конфигурации резервирования” – “Зарегистрированные области” отображен список областей, которые вы можете выбрать для резервирования. Для того, чтобы зарезервировать область, поставьте напротив её имени галочку и выполните загрузку проекта на ПЛК (см. рис. 4.14).

При необходимости резервирования нескольких программ поместите их в одну задачу в “Конфигурации задач” в дереве устройств. Синхронизация данных выбранных программ в реальном времени возможна только в пределах одной задачи – ее название прописывается в “Параметрах Резервирования” во вкладке “Общие установки”. Это означает, что если во вкладке “Зарегистрированные области” вы отметите галочкой программу, не входящую в задачу резервирования, то синхронизация данных этой программы между ПЛК происходить не будет!

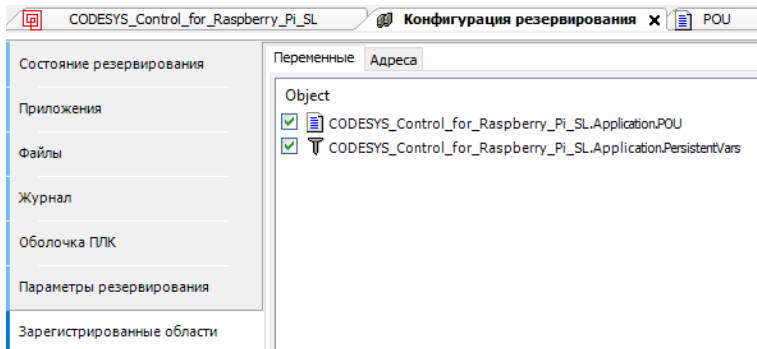


Рис. 4.14. Выбор зарегистрированных областей

При внесении изменений в проект и последующей его загрузке на ПЛК появляется окно с выбором варианта загрузки. Выберите “логин с онлайн-заменой” и установите галочку на “Обновить загрузочное приложение” или выберите “Логин с загрузкой”. (галочка установится автоматически).

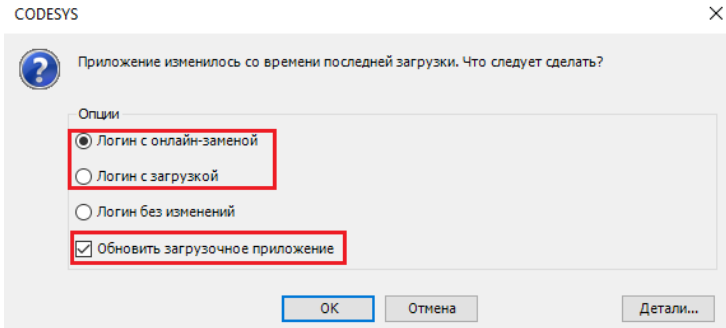


Рис. 4.15. Выбор варианта загрузки проекта на устройство

Чтобы обновление загрузочного приложения происходило всегда при онлайн-замене (галочка будет устанавливаться автоматически) нажмите правой клавишей мыши на “Application” в дереве устройств и выберите пункт “Свойства”. Перейдите во вкладку “Установки загрузочного приложения” и установите галочку на “Создавать неявное загрузочное приложение при онлайн-замене” и нажмите “Применить”.

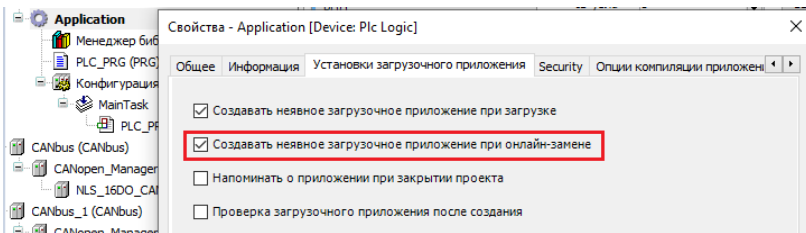


Рис. 4.16. Постоянное обновление загрузочного приложения при онлайн-замене

### 4.1.5. Функция авто-синхронизации данных между ПЛК

Для включения функции постоянной синхронизации данных между ПЛК, подключитесь к обоим ПЛК по VNC или SSH, откройте Терминал (командную строку) и введите команду:

```
sudo nano /etc/CODESYSControl_User.cfg
```



## Резервирование ПЛК в среде Codesys

Откроется редактор конфига CodesysControl. В раздел [CmpRedundancy] добавьте строку “DataSyncAlways=1”.

Для отключения достаточно закомментировать или удалить эту строку. Чтобы закомментировать строку пропишите в её начале символ « ; ».

```
[CmpRedundancy]
BootupWaitTime=5000
TcpWaitTime=2000
StandbyWaitTime=100
SyncWaitTime=0
Ethercat=0
Profibus=1
PlcIdent=1
DataSyncAlways=1
Bootproject=Application
RedundancyTaskName=MainTask
```

Рис. 4.17. Добавление функции синхронизации данных

## 4.2. Заводской сброс ПЛК в Codesys

При возникновении проблем с работой ПЛК, таких как ошибка синхронизации, ошибка запуска, ПЛК не выходит из режима эмуляции и т.д., решением является сброс обоих ПЛК к заводским настройкам.

Для того чтобы выполнить сброс ПЛК к заводским настройкам (удаление загруженного на ПЛК проекта) запустите CODESYS и откройте любой проект. Выполните сканирование устройств и подключитесь к одному из ПЛК резервной пары.

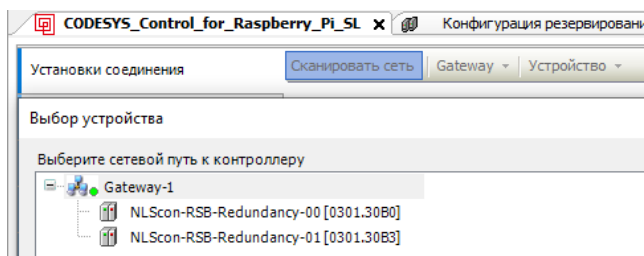


Рис. 4.18. Найденные устройства при сканировании

После того как подключение будет установлено, в дереве устройств кликните правой кнопкой мыши по названию устройства и выберите “Сброс заводской устройства” (см. рис. 4.19).

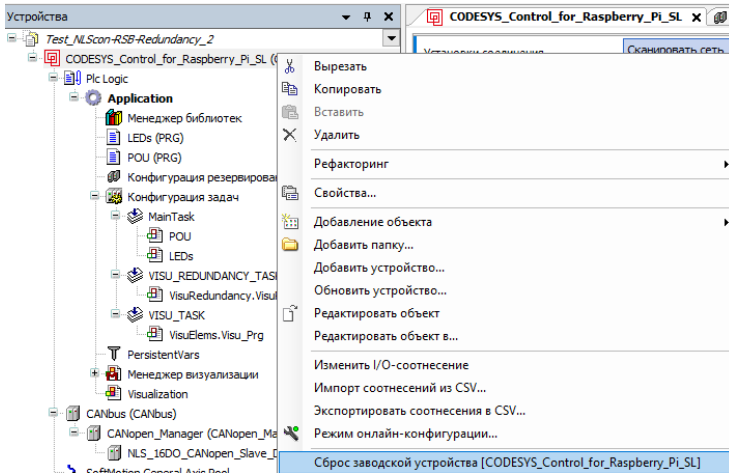


Рис. 4.19. Пункт “Заводской сброс устройства”

Во всплывающем окне с просьбой подтвердить выполнение сброса. Нажмите кнопку “Да”.

После выполнения сброса для одного из ПЛК резервной пары выполните сброс и для другого ПЛК резервной пары.

Для продолжения работы ПЛК в режиме резервирования потребуются снова выполнить запись параметров резервирования и загрузку проекта в основной ПЛК резервной пары. Более подробно этот процесс описан в п. 4.1.

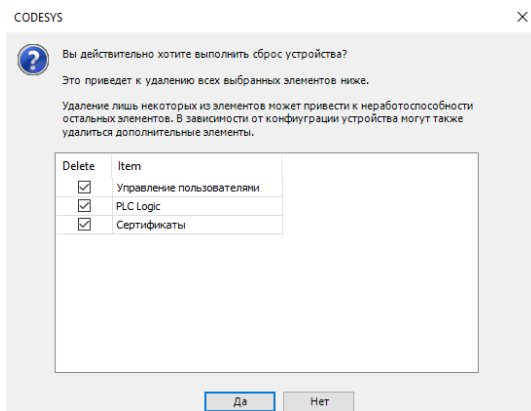


Рис. 4.20. Подтверждение сброса устройства

### 4.3. Работа переключателя Старт/Стоп

На лицевой панели ПЛК расположен переключатель Старт/Стоп, который отвечает за запуск и остановку работы МЭК-проекта CODESYS на ПЛК.

Если контроллеры используются в резервной паре и на них запущен проект, то переключатель Старт/Стоп будет работать следующим образом:

Если на **Активном** ПЛК переключатель установить в положение “Стоп”, то на обоих ПЛК работа проекта остановится. Активный станет Автономным в состоянии остановки проекта, а Пассивный перейдет в состояние Эмуляции. После возвращения переключателя в режим “Старт” работа проекта возобновится, ПЛК синхронизируются.

Если на **Пассивном** ПЛК переключатель установить в положение “Стоп”, то Пассивный остановит свою работу, а Активный станет Автономным. После возвращения переключателя в режим “Старт” на ПЛК, который был остановлен, он станет Активным. ПЛК, который находился в Автономном состоянии, синхронизируется и станет Пассивным.

Переводить переключатели обоих ПЛК одновременно в положение “Стоп” не рекомендуется. Это вызовет остановку работы МЭК-проекта на обоих ПЛК. Чтобы вернуть оба ПЛК в рабочее состояние в первую очередь переведите переключатель в положение “Старт” на Основном ПЛК. Если основной ПЛК перейдет в состояние “Автономный”, то можно переводить переключатель Резервного ПЛК в положение “Старт”. Он синхронизируется с Основным ПЛК и станет Пассивным.

## 5. Реализация дополнительных функций в CODESYS

### 5.1. Веб-визуализация

Запустите среду разработки CODESYS. Создайте стандартный проект или откройте существующий и кликните правой кнопкой мыши по “Application” в дереве устройств. Выберите “Добавление объекта – Визуализация” (см. рис. 5.1).

Во всплывающем окне с просьбой ввести имя визуализации – по умолчанию в строку имени записывается “Visualization”.

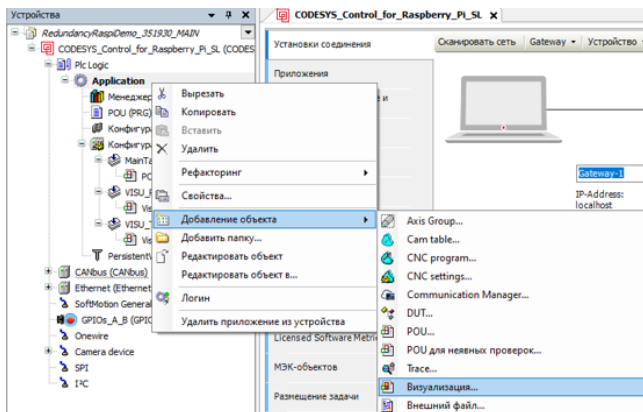


Рис. 5.1. Добавление визуализации в проект

В дереве устройств появится сама визуализация и менеджер визуализаций, где можно настроить параметры визуализации: стиль визуализации, параметры шрифта, объем памяти для визуализации и т. д. В дереве устройств у элемента “Менеджер визуализации” есть подпункт “Web-визуализация”. В данном пункте можно выбрать стартовую визуализацию, если их несколько, а также настроить масштабирование визуализации, частоту обновления и ее размер.

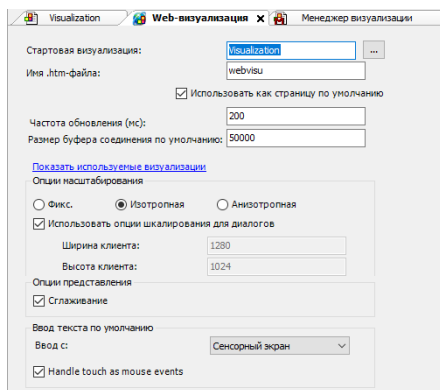


Рис. 5.2. Параметры Web-визуализации

Элементы визуализации добавляются на экран методом перетаскивания из панели инструментов визуализации, где можно выбрать категорию элементов.

Свойства элементов можно настраивать – нажатием левой кнопкой мыши на элемент, который необходимо настроить.

## Реализация дополнительных функций в CODESYS

**Например:** на экране визуализации если нажать на один из переключателей левой кнопкой мыши – с правой части экрана панель инструментов визуализации заменяется на вкладку свойств выбранного объекта. В зависимости от выбранного элемента настраиваются разные свойства. Для переключателя это размеры, переменная, которая меняет значение при его переключении и цвет.

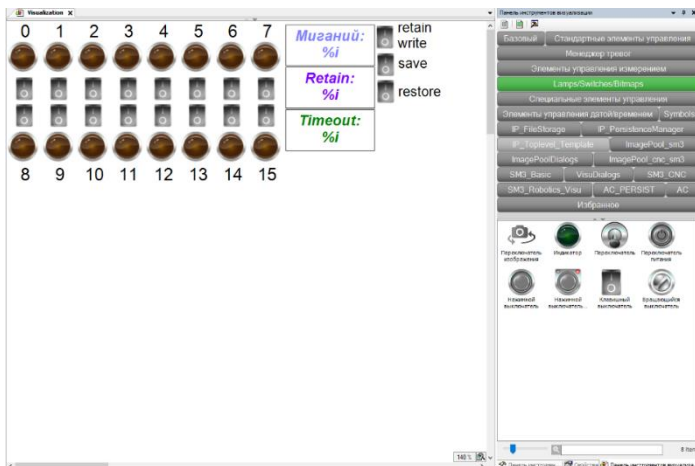


Рис. 5.3. Пример веб-визуализации

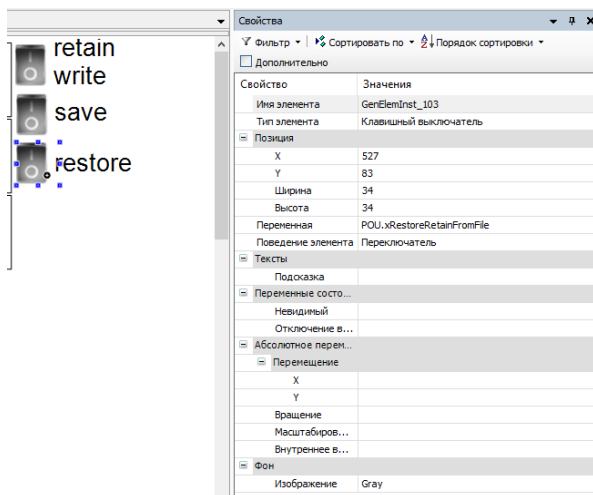


Рис. 5.4. Свойства элемента “Переключатель”

## 5.2. Работа светодиодов состояния резервирования ПЛК

Для управления светодиодами необходимо использовать шаблон проекта, встроенный в пакет с таргет-файлом. Создание проекта на основе шаблона описано в п. 4.1.1. Далее настройте параметры резервирования под ваши ПЛК и запустите проект.

Или можете создать проект с нуля без шаблона:

Создайте стандартный проект с устройством “CODESYS Control NIL AP RD”. Откройте параметры GPIO в дереве устройств.

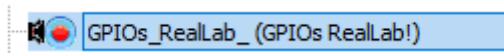


Рис. 5.5. GPIO

Тип управления GPIO8 и GPIO9 по умолчанию “Output” (см. рис. 5.6). В соотношении входов/выходов GPIO – во вкладке Output – назначить переменные на них же – GPIO8: зеленый, GPIO9: красный светодиод.

Параметр	Тип	Значение	Значение по умолчанию	Единица	Описание
GPIO0	Enumeration of BYTE	Input	Input		GPIO.0
GPIO1	Enumeration of BYTE	Output	Output		GPIO.1
GPIO2	Enumeration of BYTE	not used	not used		GPIO.2
GPIO3	Enumeration of BYTE	not used	not used		GPIO.3
GPIO8	Enumeration of BYTE	Output	Output		GPIO.8
GPIO9	Enumeration of BYTE	Output	Output		GPIO.9
GPIO12	Enumeration of BYTE	not used	not used		GPIO.12
GPIO13	Enumeration of BYTE	not used	not used		GPIO.13
GPIO30	Enumeration of BYTE	not used	not used		GPIO.30

Рис. 5.6. Установка GPIO8 и GPIO9 в состояние “Output”

	digital outputs (GPIO0..GPIO31)				
	Bit0	%QX0.0	BOOL		
	Bit1	%QX0.1	BOOL		
	Bit2	%QX0.2	BOOL		
	Bit3	%QX0.3	BOOL		
LED_GREEN	Bit8	%QX1.0	BOOL		
LED_RED	Bit9	%QX1.1	BOOL		
	Bit12	%QX1.4	BOOL		
	Bit13	%QX1.5	BOOL		
	Bit30	%QX3.6	BOOL		

Рис. 5.7. Назначение переменных в параметрах соотношения I/O

Далее создайте POU (например, с названием LEADS). Добавьте в менеджер библиотек библиотеку “CmpRedundancy Connection”.

В файле с программой LEADS объявите следующие переменные:

```
1  PROGRAM LEDs
2  VAR
3      sReduState: RedundancyState;
4      sPlcIdent: PLC_IDENT;
5      xConnectionState: BOOL;
6
7      sState: STRING;
8      TON_Standalone: TON;
9  END_VAR
10
```

Рис. 5.8. Переменные для объявления

Пример основного кода, который управляет индикацией:

```
1  GetRedundancyState(ADR(sReduState));
2  sPlcIdent := GetPlcIdent();
3  xConnectionState := GetConnectionState();
4
5  IF sReduState.eRedundancyState = RDCY.STATE.RS_CYCLE_ACTIVE THEN
6      sState := 'Active';
7      LED_GREEN := TRUE;
8      LED_RED := FALSE;
9  ELSIF sReduState.eRedundancyState = RDCY.STATE.RS_CYCLE_STANDBY THEN
10     sState := 'Passive';
11     LED_GREEN := FALSE;
12     LED_RED := TRUE;
13 END_IF
```

Рис. 5.9. Пример основного кода

### 5.3. Работа с Retain-переменными

Запустите среду разработки CODESYS. Для работы с retain-переменными, добавьте в проект PersistentVars – кликните правой кнопкой мыши по “Application” в дереве устройств, выберите “Добавление объекта – Persistent-переменные”. После добавления, кликните дважды по “PersistentVars” в дереве устройств и добавьте переменные.

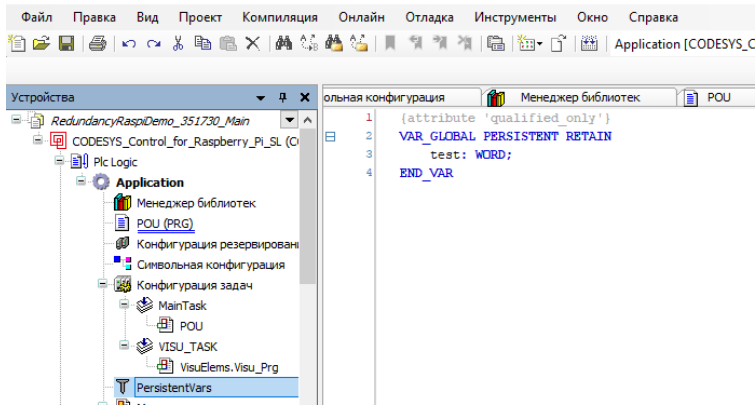


Рис. 5.10. Добавление Retain-переменных

После этого выполните загрузку проекта с обновлением загрузочного приложения на Активный ПЛК.

При потере питания от сети ПЛК сохраняет Retain-переменные и отключается через 3-5 секунд. Во время сохранения Retain-переменных ПЛК питается от встроенного аккумулятора.

### 5.4. Управление дискретными входами/выходами (GPIO)

Создайте новый проект в Codesys 3.5 на основе шаблона проекта RealLab Redundancy Project (см. п. 4.1.1) или с нуля (см. п. 4.1.2). В дереве проекта кликните дважды по пункту GPIOs\_RealLab (см. рис. 5.5) – откроется таблица конфигурации GPIO (дискретных входов/выходов).

GPIOs Конфигурация	Параметр	Тип	Значение	Значение по умолчанию	Единица	Описание
GPIOs Соотнесение входов/выходов	GPIO0	Enumeration of BYTE	Input	Input		GPIO.0
	GPIO1	Enumeration of BYTE	Output	Output		GPIO.1
GPIOs МЭК-объектов	GPIO2	Enumeration of BYTE	not used	not used		GPIO.2
	GPIO3	Enumeration of BYTE	not used	not used		GPIO.3
Состояние	GPIO8	Enumeration of BYTE	Output	Output		GPIO.8
	GPIO9	Enumeration of BYTE	Output	Output		GPIO.9
Информация	GPIO12	Enumeration of BYTE	not used	not used		GPIO.12
	GPIO13	Enumeration of BYTE	not used	not used		GPIO.13
	GPIO30	Enumeration of BYTE	not used	not used		GPIO.30

Рис. 5.11. Таблица конфигурации GPIO



## Реализация дополнительных функций в CODESYS

Табл. 3. Описание конфигурации GPIO в Codesys 3.5

GPIO	Функция	Описание
0	Output / выход	Отображает состояние зеленого светодиода “Старт”. По умолчанию функция доступа – Input. Это требуется для корректной работы именно этого светодиода. Установите тип доступа “Output” для управления состоянием светодиода.
1	Output / выход	Отображает состояние красного светодиода “Стоп”.
2	Input / Вход	Дискретный вход Din0 ПЛК.
3	Input / Вход	Дискретный вход Din1 ПЛК.
8	Output / выход	Зеленый светодиод состояния резервирования ПЛК.
9	Output / выход	Красный светодиод состояния резервирования ПЛК.
12	Output / выход	Дискретный выход Dout0.
13	Output / выход	Дискретный выход Dout1.
30	Input / Вход	Отображение питания от сети. TRUE – от сети, FALSE – от АКБ.

Чтобы выбрать тип доступа к дискретному входу/выходу, нужно выбрать Input/Output/Not used из выпадающего списка в конфигурации GPIO, столбец “значение” (см. рис. 5.12).

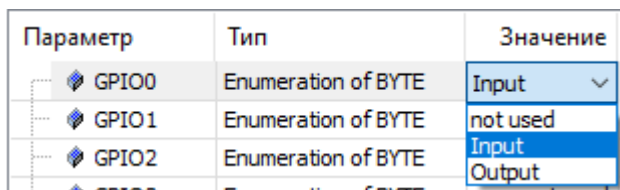


Рис. 5.12. Выбор режима доступа к GPIO

Во вкладке “Соотнесение входов/выходов” на каждый вход и выход можно назначить переменную для ее использования в программе. Для GPIO, назначенных на вход, соотносить переменную из канала digital input. Для

GPIO, назначенных на выход – соотносить переменные из канала digital outputs (см. рис. 5.13).

GPIOs Конфигурация		Найти		Фильтр	Показать все
Переменная	Соотн...	Канал	Адрес	Тип	
<b>GPIOs Соотнесение входов/выходов</b>					
GPIOs МЭК-объектов					
Состояние					
Информация					
		digital inputs (G...	%ID0	DWORD	
		Bit0	%IX0.0	BOOL	
		Bit1	%IX0.1	BOOL	
	DIN_0	Bit2	%IX0.2	BOOL	
	DIN_1	Bit3	%IX0.3	BOOL	
		Bit8	%IX1.0	BOOL	
		Bit9	%IX1.1	BOOL	
		Bit12	%IX1.4	BOOL	
		Bit13	%IX1.5	BOOL	
		Bit30	%IX3.6	BOOL	
		digital outputs (...)	%QD1	DWORD	
		Bit0	%QX4.0	BOOL	
		Bit1	%QX4.1	BOOL	
		Bit2	%QX4.2	BOOL	
		Bit3	%QX4.3	BOOL	
	LED_GREEN	Bit8	%QX5.0	BOOL	
	LED_RED	Bit9	%QX5.1	BOOL	
		Bit12	%QX5.4	BOOL	
		Bit13	%QX5.5	BOOL	
		Bit30	%QX7.6	BOOL	

Рис. 5.13. Таблица соотнесения переменных дискретных входов и выходов

После назначения соотнесения переменных с дискретными входами/выходами, можно использовать их в коде программы как логические переменные.

---

## Лист регистрации изменений

Дата изменения	Описание изменения	Примечание