



# **Среда программирования OWEN Logic**

**Руководство пользователя**

**Версия 08  
Москва  
2011**

## Содержание

Введение .....	4
Аббревиатуры и термины.....	4
1. Общие сведения о программе.....	6
1.1 Назначение .....	6
1.2 Основные характеристики .....	6
1.3 Принцип выполнения коммутационной программы.....	6
1.4 Требования к программному обеспечению и техническим средствам.....	7
2 Установка и запуск программы.....	7
2.1 Установка.....	7
2.2 Запуск.....	7
2.3 Настройка связи с ПР .....	8
3 Работа с программой .....	9
3.1 Графический интерфейс .....	9
3.2 Создание нового проекта и его сохранение .....	12
3.3 Открытие проекта для редактирования .....	13
3.4 Моделирование работы коммутационной программы.....	14
3.5 Сетевой обмен.....	15
3.6 Загрузка проекта в программируемый прибор .....	17
3.7 Дополнительные возможности .....	18
3.8 Последовательность работы над проектом.....	26
4 Быстрый старт. Создание управляющих программ .....	27
4.1 Включатель света с автоматическим отключением .....	27
4.2 Автоматическое управление электромотором мешалки .....	30
Приложение А. Функции логических элементов программы.....	34
А.1 Битовые операции .....	34
Функция «НЕ» (NOT).....	34
Функция «И» (AND).....	34
Функция «ИЛИ» (OR) .....	35
Функция «исключающее ИЛИ» (XOR) .....	35
А.2 Арифметические операции .....	35
Сложение (ADD).....	35
Сложение чисел с плавающей точкой (fADD).....	35
Вычитание (SUB).....	36
Вычитание чисел с плавающей точкой (fSUB) .....	36
Умножение (MUL).....	36
Умножение чисел с плавающей точкой (fMUL).....	37
Деление (DIV) .....	37
Деление чисел с плавающей точкой (fDIV) .....	37
Деление с остатком (MOD).....	37
Абсолютное значение числа с плавающей точкой (fABS) .....	38
Возведение в степень числа с плавающей точкой (fPOW) .....	38
А.3 Операции сравнения и выбора.....	38
Равно (EQ) .....	38
Больше (GT) .....	39
Больше, числа с плавающей точкой (fGT) .....	39
Выбор (SEL) .....	39
А.4 Операции логического битового сдвига и преобразования .....	40
Побитный логический сдвиг влево (SHL) .....	40
Побитный логический сдвиг вправо (SHR) .....	40

Чтение бита (EXTRACT).....	40
Запись бита (PUTBIT).....	41
Дешифратор (DC32) .....	41
Шифратор (CD32) .....	42
Приложение Б. Функциональные блоки программы.....	43
RS-триггер с приоритетом выключения (RS).....	43
SR-триггер с приоритетом включения (SR).....	43
Детектор переднего фронта импульса (RTRIG).....	44
Детектор заднего фронта импульса (FTRIG).....	44
D-триггер (DTRIG) .....	44
Таймер с задержкой включения (TON).....	45
Таймер с задержкой отключения (TOF).....	45
Генератор прямоугольных импульсов (BLINK) .....	46
Инкрементный счетчик с автосбросом (CT).....	46
Универсальный счетчик (CTN) .....	46
Инкрементный счетчик (CTU).....	47
Интервальный таймер (CLOCK) .....	47
Интервальный таймер с недельным циклом (CLOCK WEEK) .....	48
Лист изменений в версиях документа .....	49

## Введение

Руководство пользователя программного обеспечения OWEN Logic (ПО OWEN Logic) предназначено для ознакомления пользователя с технологией программирования коммутационных приборов, относящихся к группе программируемых реле (в частности, приборов серии ПР1хх производства ПО OWEN). Для составления программы используется визуальный язык на основе графических блоков, применяемых в цифровых электрических схемах.

Программное обеспечение OWEN Logic поставляется на компакт-диске, входящем в комплект поставки программаторов ПР-КП10 и ПР-КП20 (новые версии доступны на сайте [www.owen.ru](http://www.owen.ru)).

## Аббревиатуры и термины

Аббревиатуры и термины, используемые в тексте руководства для компактного описания:

<b>OWEN Logic</b>	– разработанная компанией ПО OWEN среда программирования.
<b>FBD (Function Block Diagram)</b>	– графический язык для визуального программирования.
<b>Коммутационная программа</b> («управляющая программа», «проект»)	– коммутационная программа, созданная пользователем в среде программирования программы OWEN Logic для последующей записи в ПР.
<b>Логический сигнал</b> («сигнал»)	– дискретная физическая величина (напряжение или ток), принимающая только два значения: включено – соответствует логической «1» (лог. «1»), и отключено – логическому «0» (лог. «0»).
<b>Макрос</b>	– функциональный блок, разработанный пользователем.
<b>ПК</b>	– персональный компьютер.
<b>ПР</b>	– программируемое реле.
<b>Прибор</b>	– программируемое устройство, например, программируемое реле ПР110.
<b>Преобразователь</b>	– устройство, через которое ПР подключается к ПК. Схема подключения преобразователя к ПР приведена в руководстве по эксплуатации на ПР.
<b>Уставка</b>	– заданное значение параметра функционального блока (установленный режим).
<b>Функциональный блок</b>	– структурная единица программы, которая после выполнения выдает одно или более значений. Может быть создано множество поименованных экземпляров (копий) функционального блока.
<b>Функция</b>	– структурная единица программы, которая после выполнения выдает только один элемент данных, который рассматривается как результат. Функция не хранит информацию о своем внутреннем состоянии, то есть вызов функции с одними и теми же фактическими параметрами выдает то же значение.

**Холст**

- поле для размещения экземпляров графических компонентов проекта и редактирования соединительных цепей между ними.

**Цикл**

- время выполнения ПР заданной коммутационной программы (зависит от количества выполняемых операций в программных цепях).

## 1. Общие сведения о программе

### 1.1 Назначение

Программное обеспечение OWEN Logic – среда программирования, предназначенная для создания алгоритмов работы приборов, относящихся к классу «программируемых реле». Эти приборы применяются для построения автоматизированных систем управления, а также для замены релейных систем защиты и контроля. При использовании ПР требуется меньше переключающих устройств для решения ряда задач малой автоматизации, что снижает затраты на проектирование и изготовление систем, а также повышает их надежность.

Перечень приборов, для программирования которых может использоваться программа, доступен на сайте [www.owen.ru](http://www.owen.ru).

### 1.2 Основные характеристики

Программное обеспечение OWEN Logic позволяет пользователю разработать коммутационную программу по собственному алгоритму с последующей записью ее в энергонезависимую память прибора.

Одновременно среда программирования может работать только с одним проектом. При создании проекта не требуется подключения ПР.

Последовательность операций при разработке проекта следующая:

- 1) установка среды программирования OWEN Logic на ПК;
- 2) запуск среды программирования;
- 3) создание нового проекта (для конкретной модели ПР) или открытие существующего проекта для редактирования;
- 4) сохранение проекта на жестком диске ПК;
- 5) отладка проекта в режиме симуляции;
- 6) загрузка проекта в ПР.

Разработка коммутационной программы в среде программирования ведется с помощью графического языка программирования FBD.

### 1.3 Принцип выполнения коммутационной программы

Коммутационная программа для ПР составляется с учетом количества имеющихся у него входов, выходов и наличия часов реального времени. Общая структура для таких программ показана на рисунке 1.1.

Работу ПР можно представить в виде последовательно выполняемых шагов:

**Шаг 1** – логическое состояние входов автоматически записывается в ячейки памяти входов (количество ячеек равно числу входов, например, I1,..., I8).

**Шаг 2** – коммутационная программа считывает значения из ячеек памяти входов и выполняет над ними логические операции, в соответствии со схемой.

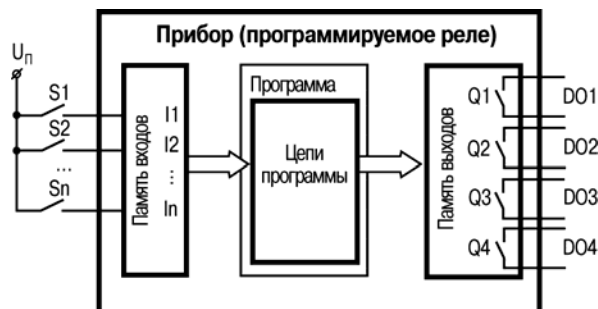


Рисунок 1.1 – Функциональная схема работы ПР,  
S1–Sn – переключатели или кнопки

**Шаг 3** – после обработки всей коммутационной программы производится запись результатов на физические выходы ПР (для включения выходных элементов Q1, ..., Qn).

**Шаг N** – переход на Шаг 1 (после выполнения всех предыдущих шагов обработки коммутационной программы цикл работы ПР повторяется с первого шага).

Время выполнения всех шагов называется рабочим циклом ПР и зависит от сложности алгоритма коммутационной программы.

## **1.4 Требования к программному обеспечению и техническим средствам**

ПО OWEN Logic работает на ПК под управлением операционных систем MS Windows 98/2000/XP/Vista. Для ее работы должна быть предустановлена среда .NET Framework 2.0. Установить .NET Framework можно с диска, поставляемого с преобразователем ПР-КП (дистрибутив поставляется с октября 2011 г), либо может быть скачан с сайта компании Microsoft.

### **Минимальные технические требования к ПК:**

- процессор Pentium 2 и выше;
- оперативная память 64 Мб и более;
- минимальное свободное пространство на жестком диске 30 Мб;
- стандартный COM-порт для работы с преобразователем ПР-КП10, либо USB-порт для работы с преобразователем ПР-КП20 (AC7);
- клавиатура и мышь;
- монитор с разрешением экрана 800х600 пикселей.

## **2 Установка и запуск программы**

### **2.1 Установка**

Для установки на компьютер программы OWEN Logic следует запустить файл Setup OWEN Logic.exe (размещен на компакт-диске преобразователя ПР-КП) и следовать инструкциям, появляющимся на экране.

После успешной установки на рабочем столе появится ярлык OWEN Logic (рисунок 2.1).



**Рисунок 2.1 –  
Ярлык программы**

### **2.2 Запуск**

Для запуска программы OWEN Logic следует выбрать команду **«Пуск ▶ Программы ▶ OWEN ▶ OWEN Logic ▶ OWEN Logic»** или дважды щелкнуть левой кнопкой мыши на соответствующей иконке рабочего стола.

### 2.3 Настройка связи с ПР

Для настройки связи с ПР сначала выполняется подключение его к ПК через любой свободный COM-порт. Для соединения используется кабель программирования, входящий в комплект поставки преобразователя ПР-КП.

**Внимание!** Электрическое подключение прибора к ПК через преобразователь ПР-КП10 допускается выполнять только при выключенном питании ПР.

После электрического соединения устройств, настраиваются параметры связи в ПО OWEN Logic.

Настройка канала соединения с ПР производится в окне «Настройка порта» – оно вызывается из основного меню командой

**Прибор ► Настройка порта....**

Диалоговое окно имеет вид, показанный на рисунке 2.2.

Настройка состоит в выборе номера используемого COM-порта (остальные настройки фиксированные и выводятся для справки).

Выбор подтверждается нажатием кнопки ОК.

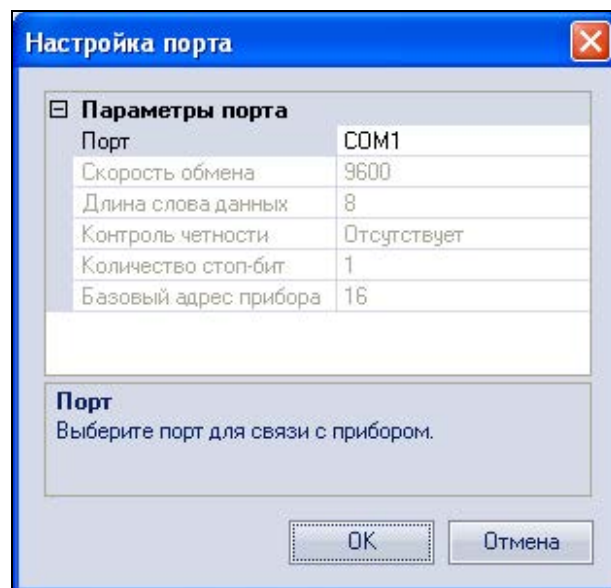


Рисунок 2.2 – Окно настройки порта



### 3 Работа с программой

#### 3.1 Графический интерфейс

После запуска ПО OWEN Logic на мониторе открывается Главное окно программы, приведенное на рисунке 3.1.

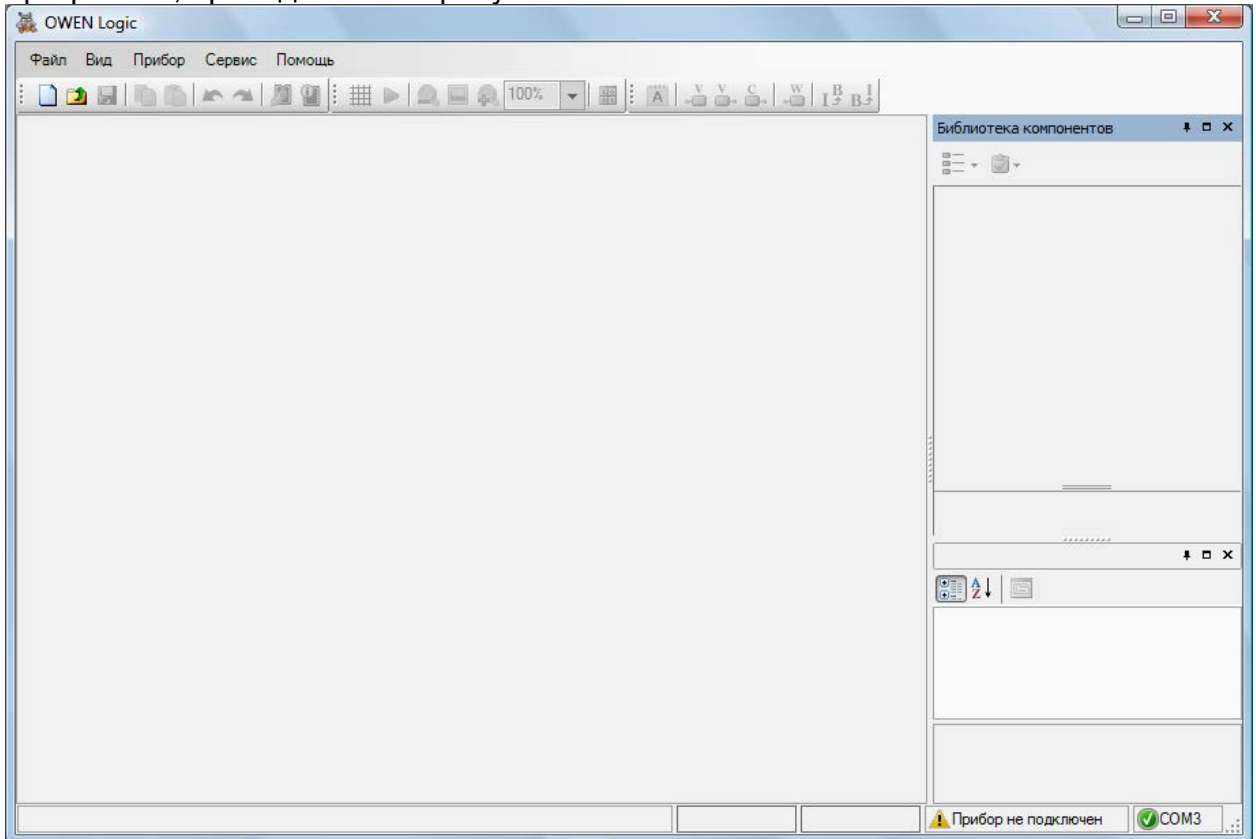






Рисунок 3.1 – Главное окно программы (вид до открытия проекта)


**Главное окно программы включает:**

- заголовок окна (верхняя строка) – после создания проекта в заголовке автоматически выводится информация об имени файла проекта и его месте размещения на ПК;
- главное меню программы: **Файл, Вид, Прибор, Сервис, Помощь**;
- панели инструментов для быстрого вызова часто используемых действий (таблица 3.1).

Таблица 3.1 – Элементы управления программы

Кнопка панели инструментов	Команда главного меню	Название и описание действий
	<b>Файл   Новый</b>	<b>Новый</b> – открывает новый проект. Текущий проект при этом закрывается
	<b>Файл   Открыть</b>	<b>Открыть</b> – открывает на редактирование ранее созданный и сохраненный проект
	<b>Файл   Сохранить</b>	<b>Сохранить</b> – сохраняет текущий проект в файле (кнопка не активна, пока проект не создан). При первоначальном сохранении вызывает окно для присвоения имени файлу
	<b>Файл   Печать...</b>	<b>Печать ...</b> – вызывает окно для печати проекта

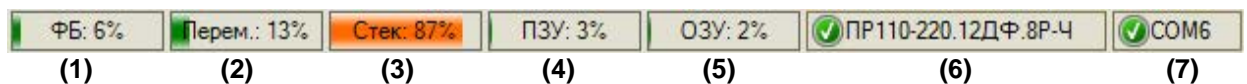
—	<b>Файл   Сохранить как...</b>	<b>Сохранить как ...</b> – вызывает окно для присвоения нового имени файлу при его сохранении
	<b>Файл   Записать программу в прибор</b>	<b>Записать программу в прибор</b> – вызывает запись созданного проекта в ПР
	<b>Файл   Информация</b>	<b>Информация</b> – вызывает информационное окно с версией программного обеспечения в ПР и целевом назначении проекта (модели ПР, для которой разработан проект)
	<b>Прибор   Таблица адресов Modbus...</b>	<b>Таблица адресов Modbus...</b> – вызывает диалоговое окно с информацией о распределении сетевых переменных в таблице Modbus
	<b>Вид   Сетка</b>	<b>Сетка</b> – вызывает отображение разметочной сетки на холсте (в зоне размещения схемы проекта)
	—	<b>Режим симулятора</b> – запуск режима моделирования работы коммутационной программы
	—	<b>Уменьшение масштаба</b> – изменяет масштаб рабочего поля схемы проекта
	—	<b>Оригинальный размер</b> – возвращает к масштабу рабочего поля схемы проекта, установленному по умолчанию
	—	<b>Увеличить масштаб</b> – изменяет масштаб рабочего поля схемы проекта
	—	<b>Масштаб рабочей зоны</b> (холста) при отображении схемы проекта
	<b>Вид   Переименовать компоненты</b>	<b>Переименовать компоненты</b> – изменяет нумерацию функциональных блоков, расположенных в рабочей зоне проекта (кнопка не активна, пока проект не создан)
	—	<b>Блок комментариев</b> – обеспечивает добавление комментариев на схему проекта
	—	<b>Создание выходной переменной</b> – обеспечивает удобство выполнения связей на схеме (без указания соединительной линии)
	—	<b>Создание входной переменной</b> – обеспечивает удобство выполнения связей на схеме (без указания соединительной линии)
	—	<b>Создание блока константы</b> – обеспечивает удобство задания фиксированных числовых значений (пороговых или уставок) для ФБ, работающих с целочисленными значениями
	—	<b>Создание входной сетевой переменной</b> – создание переменной, значение которой задается по сети.

	–	<b>Создание выходной сетевой переменной</b> – создание переменной, значение которой считывается по сети.
	–	<b>Создание блока записи в ФБ</b> – обеспечивает подачу значений переменной на вход ФБ, при этом на схеме может не отображаться соединительная линия цепи передачи сигнала
	–	<b>Преобразователь в булевское значение</b> – преобразовывает целочисленное значение сигнала в булевское
	–	<b>Преобразователь в целочисленное значение</b> – преобразовывает булевское значение сигнала в целочисленное

– вкладки «Библиотека компонентов» и «Свойства» (до открытия проекта в них нет информации);

– поле редактирования создаваемой коммутационной программы – рабочая область проекта (до открытия проекта холст пустой);

– нижнюю статусную строку, показывающую состояние программы OWEN Logic и наличие подключения к ней ПР (рисунок 3.2).




**Рисунок 3.2 – Статусная строка окна программы OWEN Logic при подключенном приборе**

В строке статуса отображается информация:

- (1) – доступное количество экземпляров функциональных блоков;
- (2) – доступное количество переменных. Переменные создаются неявным образом в «обратных связях», во множественных связях, где есть общий «узел»;
- (3) – доступное количество уровня стека;
- (4) и (5) – доступная память ПР в процентах от общего объема: «ПЗУ: 0...100 %» и «ОЗУ: 0...100 %» (чем больше используется в программе функциональных блоков и функций, тем больше памяти требуется для работы в ПР). ПО OWEN Logic автоматически рассчитывает доступную память ПР, и, в случае критического значения, выводит соответствующее предупреждение;
- (6) – информация о наличии связи между ПО OWEN Logic и ПР, при наличии связи указывается модель подключенного ПР;
- (7) – номер выбранного пользователем порта.

После создания проекта появится доступ к вкладкам «Функция», «Функциональный блок» и «Макрос», на которых размещены компоненты, используемые при составлении программы. Вкладка «Макросы» позволяет расширить число применяемых в проекте функциональных блоков за счет созданных пользователем самостоятельно.

### 3.2 Создание нового проекта и его сохранение

После запуска ПО OWEN Logic следует нажать кнопку «Создать» (  ) панели инструментов – при этом появится окно выбора типа ПР, для которого будет создаваться проект (рисунок 3.3).

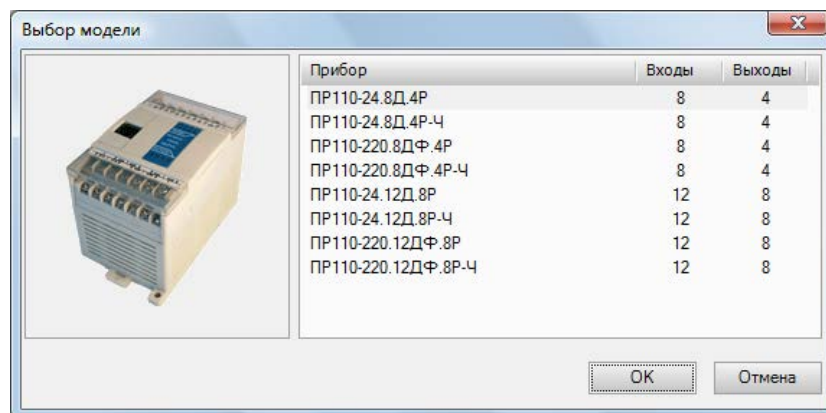


Рисунок 3.3 – Окно «Выбор модели» ПР для проекта

После выделения курсором нужной модели ПР, выбор подтверждается нажатием кнопки ОК, – в рабочей области программы появится поле (холст), на краях которого слева размещены входы «I», а справа – выходы «Q» ПР (рисунок 3.4).

Проект создается путем перетаскивания выделенных курсором мыши блоков из вкладки «Библиотека компонентов» на рабочее поле (холст) и виртуального соединения цепей блоков между собой и с входами и выходами ПР. Для установки соединений цепей блоков пользователь курсором мыши указывает начальную и конечную точки привязки.

**Примечание** - Связь не может устанавливаться только между входами (или только между выходами) графических компонентов, – допустимым является соединение выхода и входов.

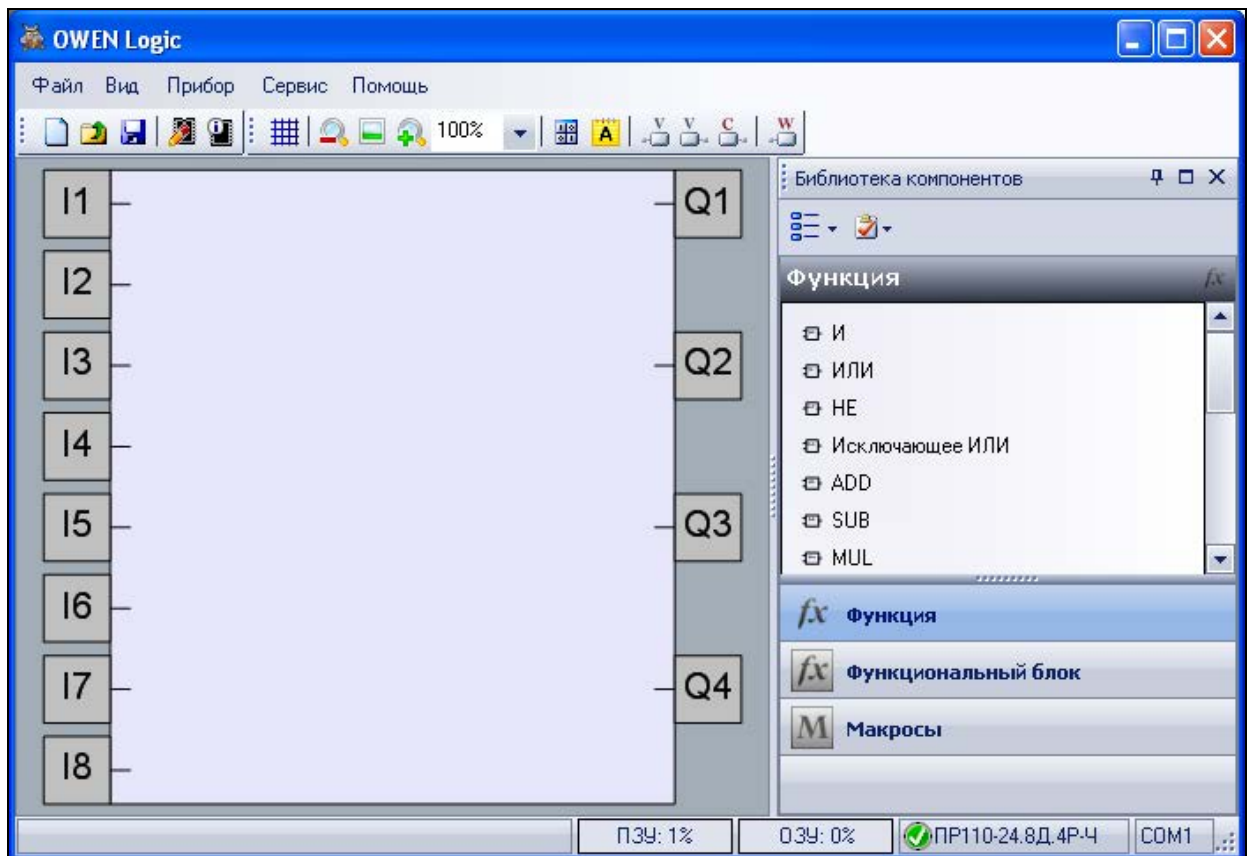



Рисунок 3.4 – Вид окна программы после выбора модели ПР (прибор подключен)

Для присвоения проекту идентификационного имени следует нажать кнопку «Сохранить проект» (  ) панели инструментов, или выбрать команду «Файл ► Сохранить как...» – откроется окно «Сохранить как» в котором можно присвоить имя файлу (в поле «Имя файла:», рисунок 3.5).

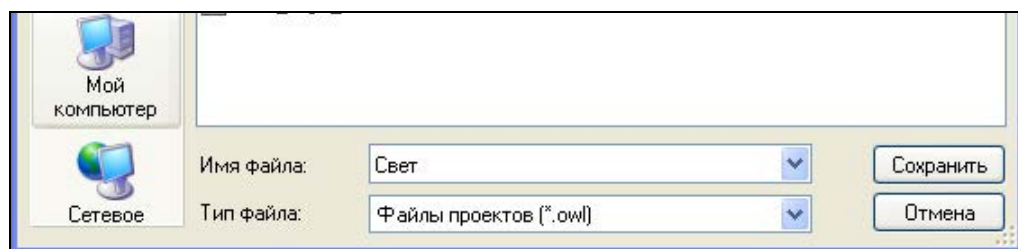



Рисунок 3.5 – Фрагмент окна «Сохранить как»

При первом сохранении или при переименовании файла проекта – в открывшемся стандартном окне выбора файла – следует выбрать либо задать наименование и расположение файла.

Расширение файла проекта: «\*.owl». Таким образом, название файла проекта может выглядеть, например, «Свет\_Logic.owl».

### 3.3 Открытие проекта для редактирования

Для того чтобы внести изменения в проект, ранее сохраненный в файл, этот файл следует открыть из ПО OWEN Logic.

Для того чтобы открыть требуемый файл, следует нажать кнопку «Открыть» (  ) панели инструментов, или выбрать команду «Файл ► Открыть» главного меню. В открывшемся стандартном окне выбора файла – выбрать требуемый

файл. Текущий проект будет закрыт, а в окне программы – открыт выбранный проект.

### 3.4 Моделирование работы коммутационной программы

Проверка корректности работы созданной коммутационной программы проводится с помощью режима симуляции, в котором моделируется изменение состояний выходов в зависимости от изменения состояний входов. Кроме того, моделирование позволяет проанализировать состояние сигналов внутри коммутационной программы.


Для перехода в режим симуляции необходимо воспользоваться кнопкой «Режим симулятора» (). После перехода в режим создается панель (рисунок 3.7).



Рисунок 3.7 – Панель симулятора

На панели расположены следующие элементы управления:

- кнопка (1) для запуска моделирования в реальном времени;
- кнопка (2) для пошагового моделирования;
- кнопка (3) для приостановления работы симулятора, запущенного в реальном времени;
- кнопка (4) для остановки работы симулятора – перевод режима симуляции в первоначальное состояние;
- поле ввода (5) для задания времени периода обновления информации на схеме;
- поле ввода (6) для задания времени цикла для временных функциональных блоков – «TON», «TOF», «BLINK».

Для вариантов исполнений приборов с функцией часов реального времени создается дополнительная панель, при помощи которой можно моделировать поведение функциональных блоков «CLOCK» и «CLOCK WEEK» во времени (рисунок 3.8).

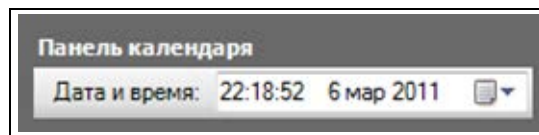




Рисунок 3.8 – Панель календаря

После перехода в режим симуляции, моделирование коммутационной программы, в общем случае, происходит в следующей последовательности:

- 1) запуск симулятора в одном из режимов – «режим реального времени» () или «пошаговый режим» ();
- 2) задание значений входных сигналов, используя нажатие на блоке входов (рисунок 3.9);

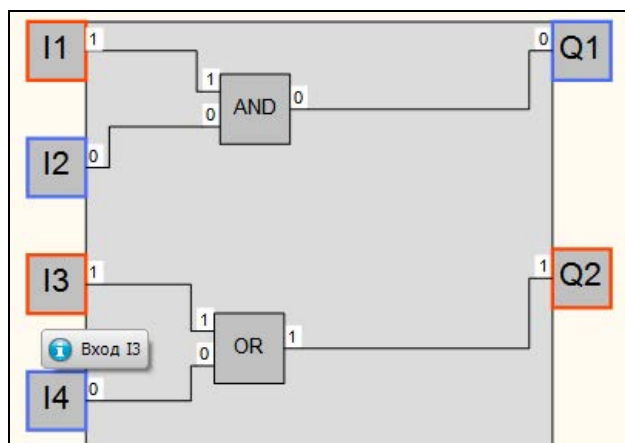


Рисунок 3.9 – Пример моделирования коммутационной программы

3) подбор значений параметров «период, мс» и «время цикла, мс» для удобства моделирования;

4) выход из режима симуляции для корректировки коммутационной программы.

**Примечание.** Моделирование основной коммутационной программы (вкладка «Схема») и макросов производится в отдельности.

## 3.5 Сетевой обмен

### 3.5.1 Работа с сетевыми переменными

Для реализации сетевых возможностей приборов ПР110, в среде программирования предусмотрен специальный тип переменных – сетевые переменные, рисунок 3.10.





Рисунок 3.10 – Сетевые переменные

Назначение каждой сетевой переменной определяется самим пользователем на этапе разработки коммутационной схемы. Например, сетевая переменная может служить для задания по сети уставки функционального блока «Таймер с задержкой включения», «Счётчика» и др., или сетевая переменная может быть применена для считывания текущего состояния выхода функционального блока «Универсальных счётчик».


Переменные, значение которых передаётся по сети в программу пользователя, называются сетевые входы. Переменные, которые могут быть считаны по сети, – сетевые выходы.

Добавление сетевых переменных в проект рекомендуется выполнять в следующем порядке:

**Шаг 1** – щелкнуть мышкой по кнопке  или по кнопке  на панели инструментов, затем установить курсор на место, где должна располагаться переменная и еще раз нажать левую кнопку мыши – на холсте появится блок сетевой переменной.

**Шаг 2** – курсором выделяется элемент сетевой переменной и на



соответствующей ему закладке «Свойства» в поле ввода «Адрес переменной» нажимается кнопка , рисунок 3.11.

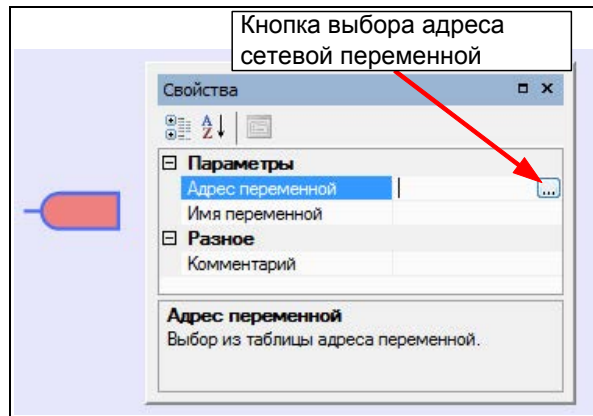


Рисунок 3.11 – Выбор адреса сетевой переменной

**Шаг 3** – в появившемся диалоговом окне выбирается адрес сетевой переменной и нажимается кнопка «ОК». Также в данном окне можно ввести имя и комментарий к сетевой переменной, рисунок 3.12.

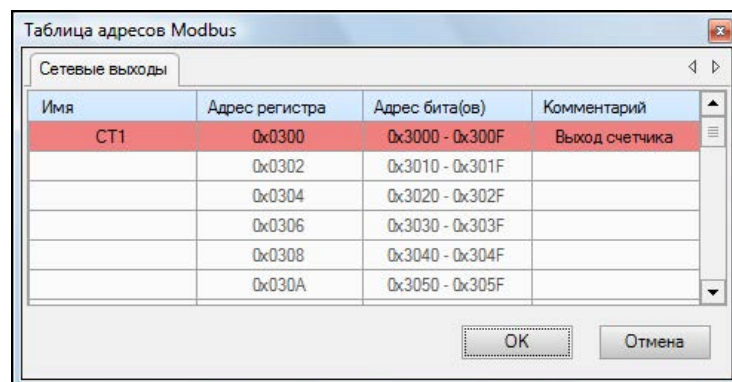


Рисунок 3.12 – Выбор адреса сетевой переменной в таблице Modbus

**Шаг 4** – подсоединить выход/вход сетевой переменной к необходимому компоненту, рисунок 3.13;

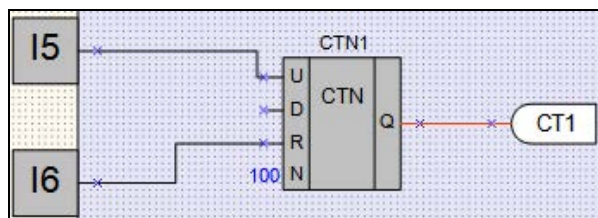


Рисунок 3.13 – Пример применения выходной сетевой переменной: чтение по сети текущего значения выхода ФБ «Универсальный счетчик»

Вышеописанный алгоритм работы с сетевыми переменными удобен, если пользователь начинает работу с проектом с программирования прибора ПР. В случае, если список параметров Мастера сети Modbus уже известен, работу удобнее начинать с определения сетевых переменных в таблице адресов Modbus. Вызов таблицы адресов Modbus – рисунок 3.14.



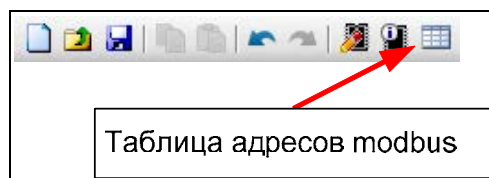


Рисунок 3.14 – Таблица адресов Modbus

### 3.5.2 Конфигурирование параметров интерфейса связи RS-485

Задание параметров интерфейса связи прибора ПР осуществляется через диалоговое окно «Настройка сетевых параметров RS-485», выбрав команду «**Прибор ► Сетевые параметры RS-485...**» главного меню, при подключенном приборе – рисунок 3.15.

Параметры вступают в силу после перезагрузки питания прибора ПР110 или после загрузки проекта в прибор.

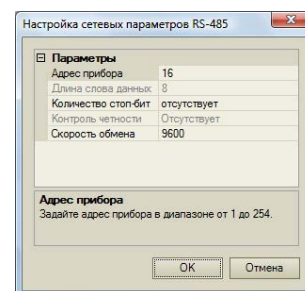



Рисунок 3.15 – Конфигурирование сетевых параметров RS-485

## 3.6 Загрузка проекта в программируемый прибор

После создания проекта его записывают в энергонезависимую память ПР. Для записи необходимо:

- 1) соединить ПР с COM-портом ПК при помощи коммуникационного кабеля;
- 2) подать питание на ПР согласно его руководству по эксплуатации;
- 3) настроить параметры соединения, если в этом есть необходимость (см. п. 2.3);

4) записать проект в ПР при помощи кнопки  панели инструментов, или выбрать команду «**Файл ► Записать программу в прибор**».

**Примечание.** Сразу после записи ПР переходит в рабочий режим и коммутационная программа запускается автоматически.

**Внимание!** Если в подключенном ПР уже есть ранее записанная коммутационная программа, то при записи в него нового проекта она заменяется.

### 3.7 Дополнительные возможности

Для удобства работы над проектом в программе предусмотрены дополнительные возможности:

1) **форму отображения доступных функций и функциональных блоков** для вкладок «Библиотеки компонентов» допускается изменять, рисунок 3.16.

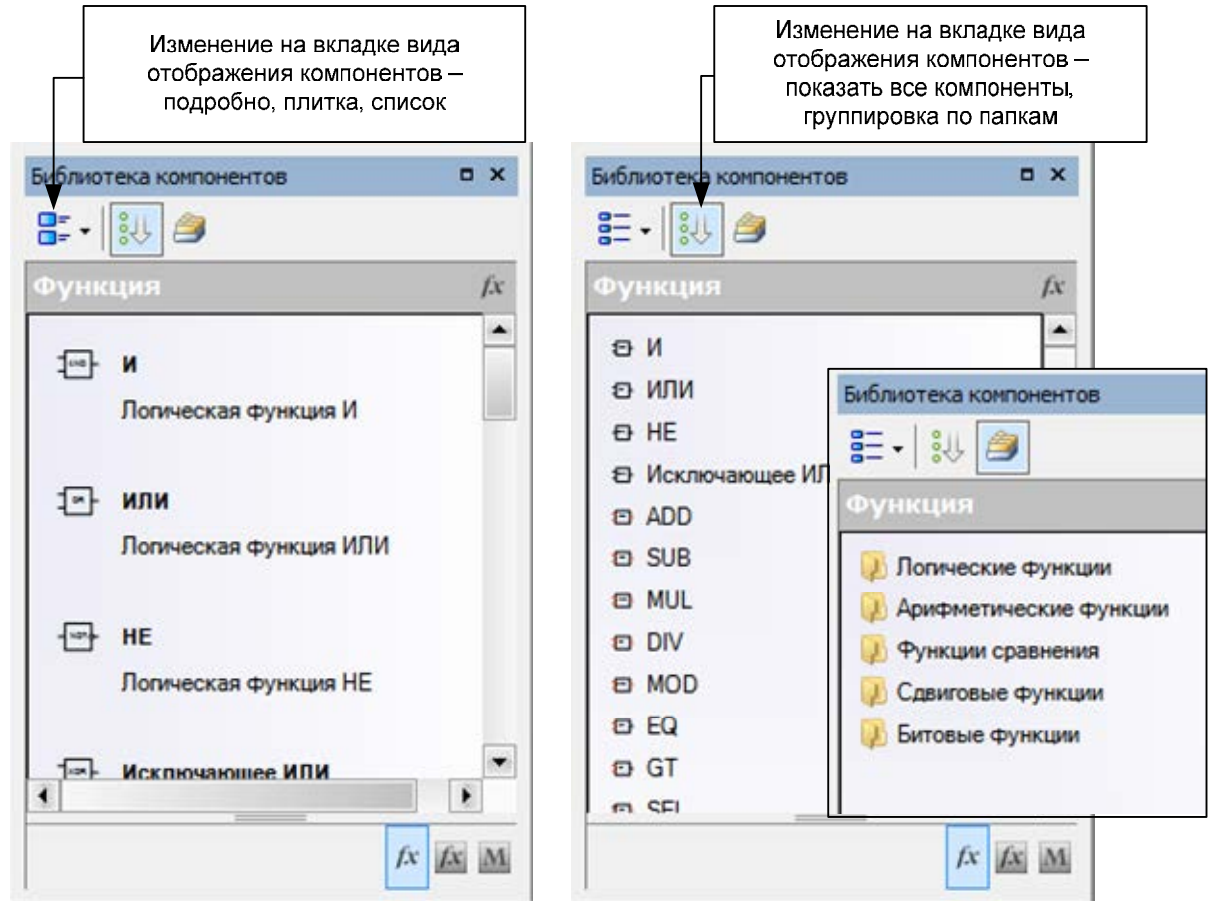




Рисунок 3.16 – Пример изменения отображения блоков библиотеки на вкладке «Функция»

Аналогичные возможности по изменению порядка (вида) отображаемой информации предусмотрены и для вкладки «Свойства»;

2) **зону отображения на вкладках**, при помощи манипулятора «мыши», пользователь может увеличивать (растягивать), переносить в любое место экрана или закрывать вкладку (убирать с экрана);

3) **масштаб рабочей зоны холста (рабочего поля проекта)** пользователь может ступенчато изменять при помощи кнопок на панели инструментов (  и  );

4) **размеры холста (рабочего поля проекта) и шаг сетки (по вертикали и горизонтали)** пользователь может изменять для удобства размещения всех компонентов проекта. Необходимые цифры размеров указываются на закладке «Свойства», рисунок 3.17;

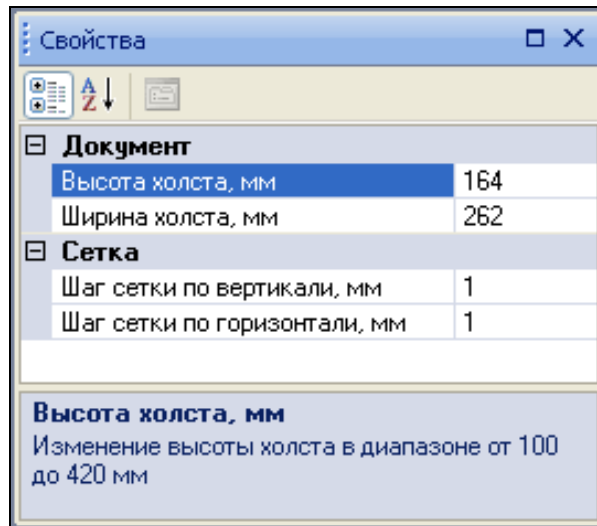



Рисунок 3.17 – Закладка «Свойства» с параметрами по умолчанию

5) **порядковые номера функциональных блоков схемы** пользователь может автоматически переименовывать при помощи кнопки () на панели инструментов;

6) **скрытые комментарии к компонентам схемы** пользователь может добавлять на закладке «Свойства», – они будут отображаться в виде всплывающей подсказки только при установке курсора на компонент;


7) **видимые комментарии на схему проекта**, добавляемые пользователем, могут выделяться на схеме цветом. Чтобы поместить на страницу какой-либо текст, следует щелкнуть мышкой (т. е. кратковременно нажать правую кнопку мыши) по кнопке () на панели инструментов, затем установить курсор на место, где должна располагаться надпись и при нажатой правой кнопке провести по диагонали для образования прямоугольника. В результате в указанном месте появляется прямоугольная рамка текстовой области с приглашением программы ввести надпись (рисунок 3.18).



Рисунок 3.18 – Сектор холста для внесения комментариев


Изменить размеры текстовой области (рамки) можно с помощью мыши – перетаскиванием. Если текст не будет целиком помещаться, он обрезается по границам рамки.

Чтобы задать текст надписи, следует щелкнуть мышью в зоне рамки и с клавиатуры ввести текст. Настроить шрифт, цвет надписи и цвет фона можно на панели вкладки «Свойства».

Текстовую область можно также переместить в необходимое положение на холсте;

8) **соединительные линии** – все входы и выхода компонентов схемы соединяются линиями (при помощи курсора мыши). После соединения цепей линиями, путь линии можно передвинуть курсором мыши при нажатой правой кнопке, установив курсор в зоне линии;

9) **изменение положения входов и выходов на холсте рабочего поля** допускается выполнять для удобства отображения схемы – все квадраты входов и выходов перемещаются (курсором мыши) в вертикальной плоскости;

10) **информация о подключенном типе ПР и версии программного обеспечения** может быть получена при помощи кнопки на панели инструментов () , рисунок 3.19;

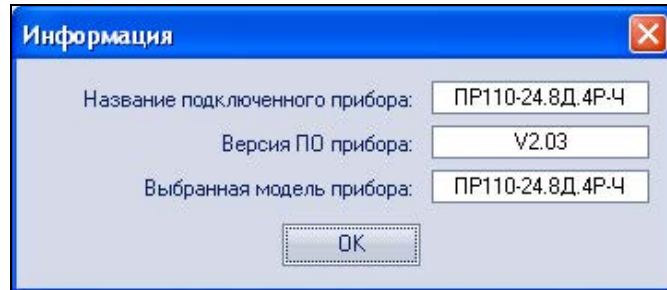


Рисунок 3.19 – Экранная форма с информацией о ПР

11) **параметры работы функций и функциональных блоков** пользователь может изменить на закладке «Свойства», доступ к которой появляется после выделения компонента схемы, если это предусмотрено в его назначении. Доступ к изменениям параметров на закладке «Свойства» появляется при выделении курсором соответствующей строки в окне;

12) **настройки часов реального времени** могут быть произведены через пункт меню «Прибор» ► «Чтение/запись времени...», – активируется только при подключении к ПК варианта исполнения ПР со встроенными часами.

Для более точного хода часов предусмотрена возможность автоматической коррекции значений, для чего в поле «Отклонение» следует задавать разницу между реальным временем и временем ухода часов прибора. Например, если ожидаемое время часов прибора после месяца работы 10:00:00, а прочитанное составило 09:59:45 (часы отстают на 15 сек), в этом случае пользователю необходимо в поле ввода «Коррекция ► Отклонение» (рисунок 3.20) установить значение «-15» сек/мес.

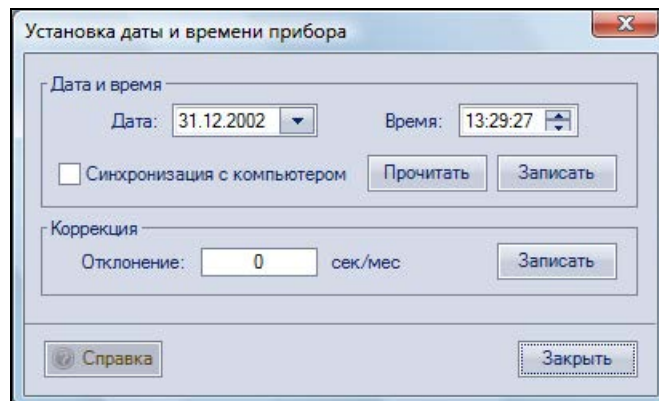




Рисунок 3.20 – Экранная форма «Установка даты и времени прибора»

13) **промежуточные выходные и входные переменные** () обеспечивающие передачу сигнала из одной цепи (например, с выхода блока или входа ПР) в другую цепь (на вход блока или выход ПР), используются без

отображения соединительной линии (связь указывается в свойствах переменной). Цепи соединений показываются только от элементов переменных, т. е. откуда сигнал идет или куда поступает дальше.

Добавление промежуточных переменных в проект рекомендуется выполнять в следующем порядке:

**Шаг 1** – щелкнуть мышкой по кнопке  на панели инструментов, затем установить курсор на место, где должна располагаться входная переменная и еще раз нажать левую кнопку мыши – на холсте появится блок входной переменной, рисунок 3.21.

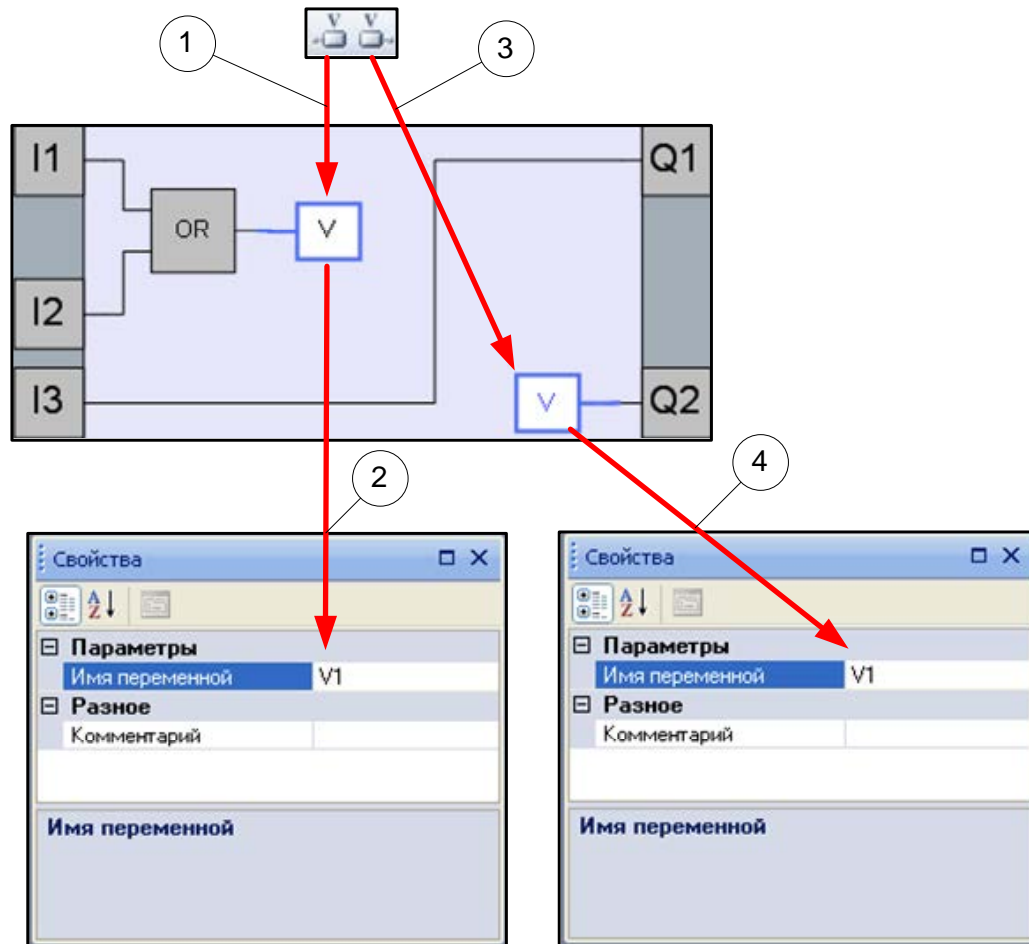




Рисунок 3.21 – Порядок действий при настройке параметров функционального блока


**Шаг 2** – курсором выделяется элемент входной переменной и на соответствующей ему закладке «Свойства» в поле ввода «Имя переменной» вводится произвольное имя (например, V1). Имя отобразится внутри блока.

**Шаг 3** – щелкнуть мышкой по кнопке  на панели инструментов, затем установить курсор на место, где должна располагаться выходная переменная и еще раз нажать левую кнопку мыши – на холсте появится блок выходной переменной.

**Шаг 4** – курсором выделяется блок выходной переменной и на соответствующей ему закладке «Свойства» в поле ввода «Имя переменной» выбирается имя переменной, из которой будет поступать сигнал – в данном случае это имя V1;

14) **блок ввода констант** создается пользователем при помощи кнопки  и применяется для подачи фиксированных значений (уставок) на входы блоков

схемы, если это необходимо для их работы. Допустимый диапазон значений зависит от типа блока, см. Приложения А и Б;

15) **задание уставки для блока из программы** выполняется пользователем при помощи кнопки . Параметры этого компонента – ссылка на уставку ФБ программы пользователя.

Например, на рисунке 3.22 показано использование компонента «Создание блока записи в ФБ» для передачи значения уставки на вход **TI** блока **BLINK1**.

**Внимание!** Значения уставок, установленные при помощи вкладки «Свойства», будут проигнорированы.

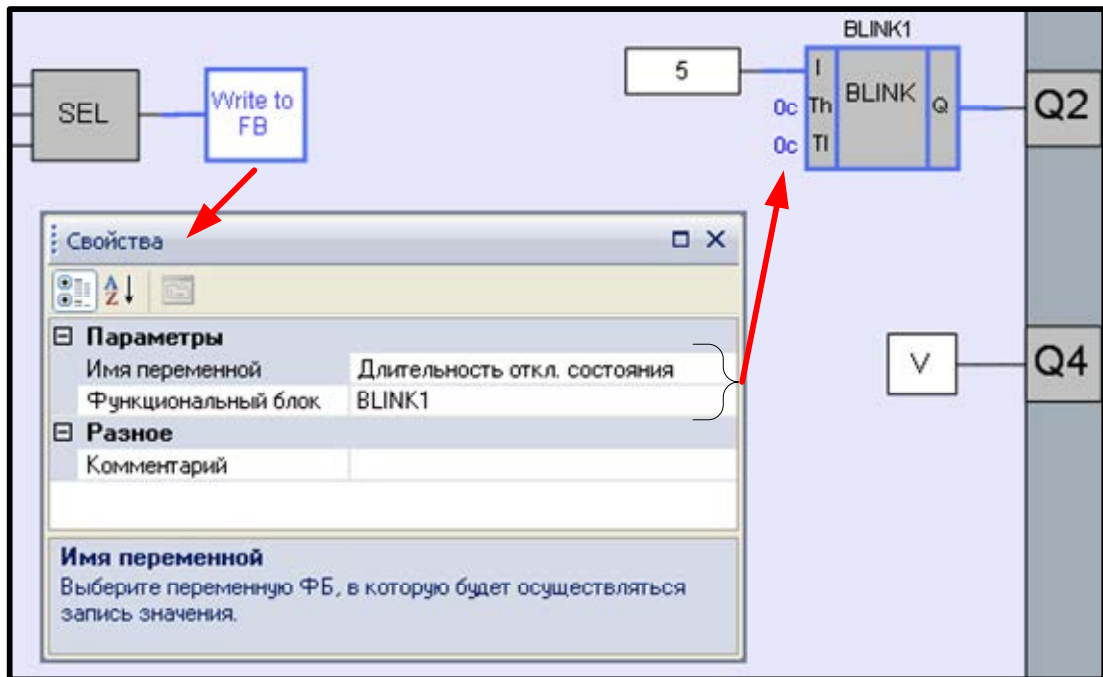


Рисунок 3.22 – Установка свойств для компонента «Создание блока записи в ФБ (Write to FB)»

Для ФБ таймеров и генераторов, где формат уставки – время, блок «Write to FB» передает значение в формате миллисекунд (см. рисунок 3.23).

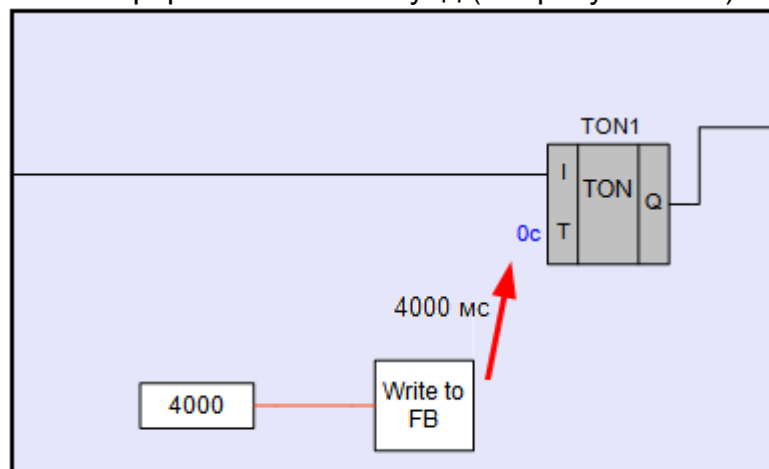


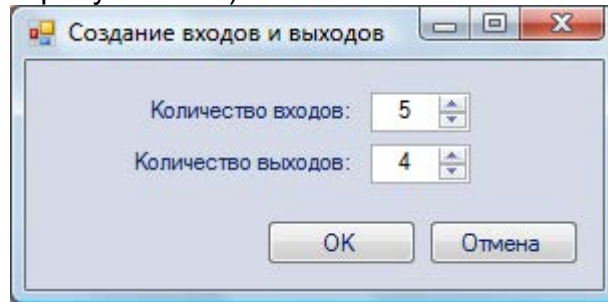
Рисунок 3.23 – Пример записи уставки для ФБ «TON»

16) **работа с макросами** выполняется в следующей последовательности:

а) создание макроса при помощи выбора команды «**Файл ► Создать макрос**». В появившемся диалоговом окне необходимо задать количество входов



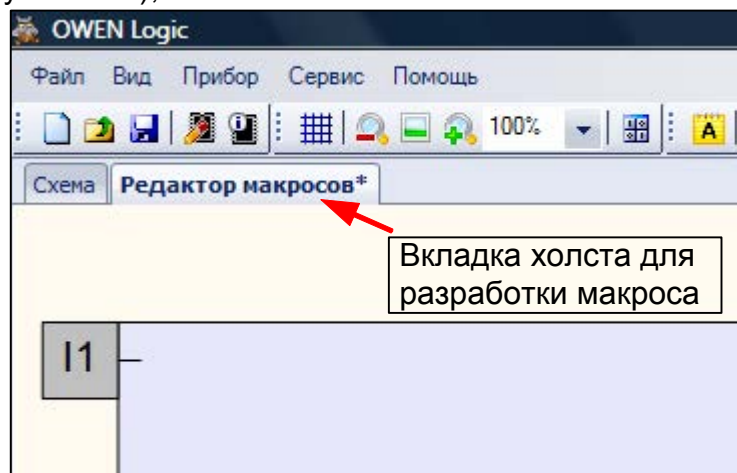
и выходов в макросе (см. рисунок 3.24).



**Рисунок 3.24 – Диалоговое окно задания количества входов и выходов макроса**

**Примечание** - В текущей версии ПО OWEN Logic после создания макроса невозможно изменить количество входов и выходов.

б) разработка алгоритма работы макроса ведется на вкладке «Редактор макросов» (см. рисунок 3.25);



**Рисунок 3.25 – Вкладка «Редактор макросов»**

Разработка алгоритма выполняется аналогичным, как и разработка основной коммутационной программы, образом.

в) задание имени и описания для макроса (см. рисунок 3.26);

г) определение типов и имен входов/выходов (см. рисунок 3.27);

д) задание «видимости» из коммутационной программы уставок функциональных блоков (см. рисунок 3.28).

При задании параметра «Использование в макросе», уставки выбранного ФБ становятся доступны при использовании ФБ в коммутационной программе аналогично уставкам ФБ TON, TOF и т.п.

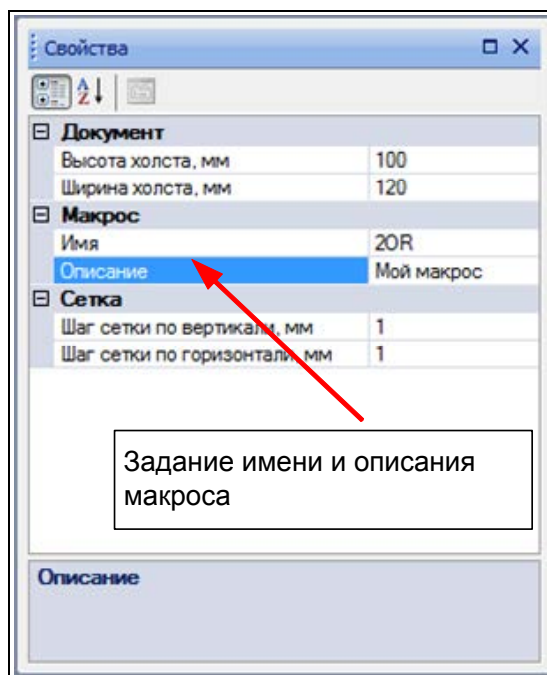


Рисунок 3.26 – Задание имени и описания макроса

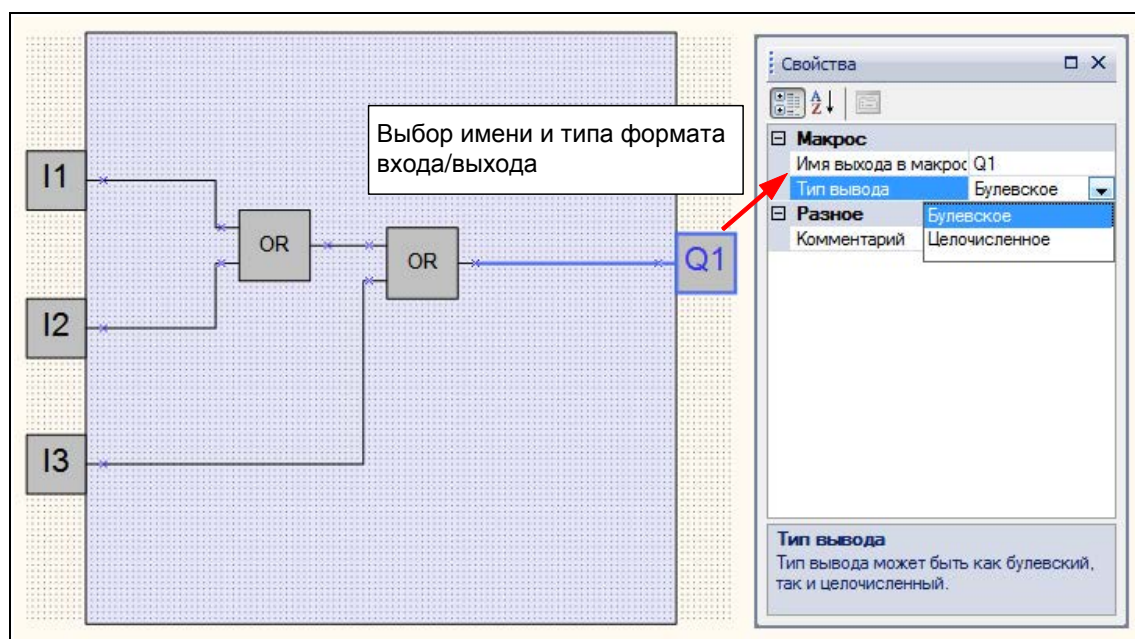


Рисунок 3.27 – Пример задания имени и типа выхода



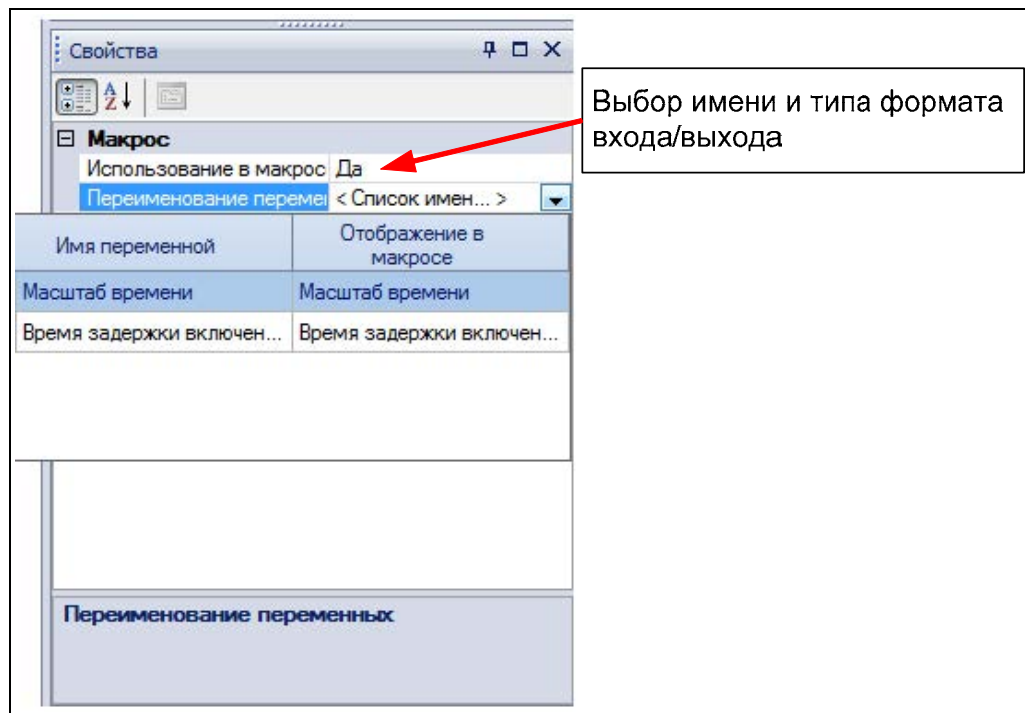


Рисунок 3.28 – Задание «видимости» уставок в коммутационной программе

е) сохранение макроса в проекте, используя команду меню **«Файл ► Сохранить макрос»** или нажав комбинацию клавиш **«Ctrl+Alt+S»**;

ж) использование макроса в коммутационной программе выполняется выбором вкладки «Макросы» на панели «Библиотека компонентов» (см. рисунок 3.29) и перенесением его на холст основной коммутационной программы.

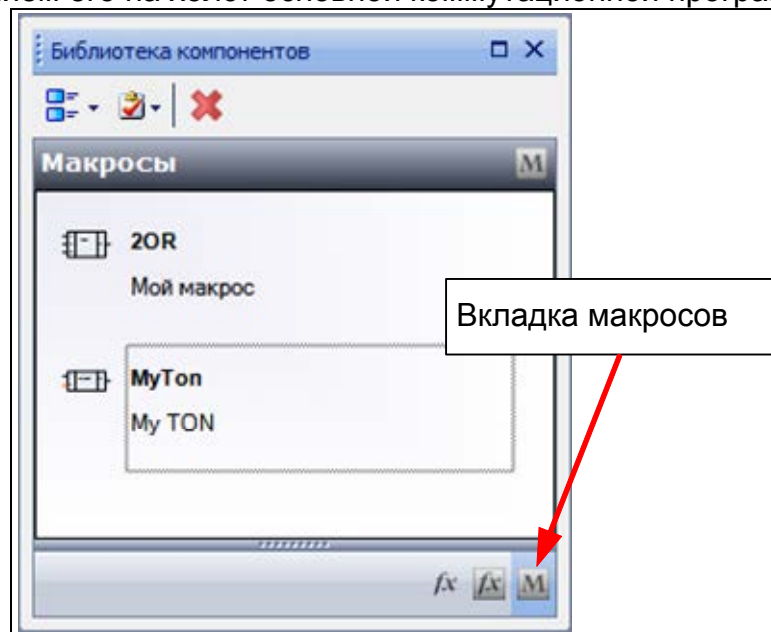


Рисунок 3.29 – Вкладка «Макросы»

Вкладка «Макросы» доступна только тогда, когда вкладка «Схема» активна.

Существуют дополнительные функции работы с макросами: «Импорт макроса из файла» и «Экспорт макроса в файл». Используя данные функции, имеется возможность сохранить макросы на жестком диске ПК с последующим использованием их в других проектах.

Работа с данными функциями ведется с помощью команд меню **«Файл ► Импорт макроса из файла»** и **«Файл ► Экспорт макроса в файл»**.

### 3.8 Последовательность работы над проектом

Составление коммутационной программы рекомендуется начинать с планирования. План должен описывать все возможные состояния ПР при функционировании (в виде диаграммы режимов, таблицы состояний, электрической или функциональной схемы и/или др.).

После того как продуманы все задачи, которые должны выполняться, необходимо составить программу на основе функций (логических элементов) и функциональных блоков, приведенных в Приложениях А и Б.

Работа над проектом включает:

1) открытие нового проекта – весь проект будет храниться в одном файле, которому следует присвоить идентификационное имя (см. п. 3.2);

2) формирование структуры текущего проекта рекомендуется выполнять в следующем порядке:

а) из «Библиотека компонентов» на холст добавляются необходимые блоки путем перетаскивания их мышью при нажатой на ней левой кнопке (из соответствующей вкладки «Функции» или «Функциональные блоки»);

б) последовательно выделяя курсором блоки схемы, на закладке «Свойства» установить их параметры, например, как это показано на рисунке 3.30;

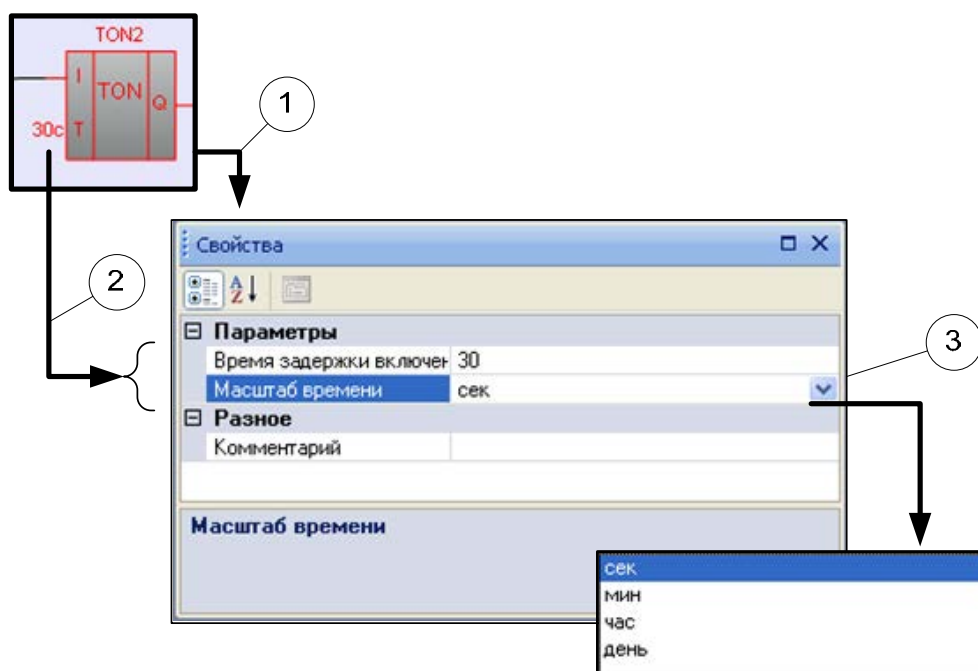


Рисунок 3.30 – Порядок действий при настройке параметров функционального блока

в) соединение компонентов программы между собой, а также с нужными входами и выходами ПР. При этом допускается передвигать квадраты входов и выходов в вертикальной плоскости для расположения соединительных линий по кратчайшей длине;

3) моделирование работы коммутационной программы в режиме симуляции. При проверке правильности работы коммутационной программы изменяют состояние входов, контролируя состояние выходов на соответствие нужным условиям;

4) загрузка проекта в ПР и проверка его работы.

5) после устранения всех ошибок подготовка проекта завершается сохранением его в файле.

Процесс создания коммутационной управляющей программы иллюстрируется на конкретных примерах, приведенных в разделе 4.

## 4 Быстрый старт. Создание управляющих программ

В данном разделе на примере простых задач поясняется, каким образом составляются управляющие коммутационные программы в среде программирования OWEN Logic. Все приведенные примеры имеются на компакт-диске (в папке «Примеры для ПР110»), поставляемом с программой OWEN Logic.

### 4.1 Включатель света с автоматическим отключением

В подсобных помещениях и коридорах электрический свет нужен ограниченное время. После включения освещения, его нередко забывают отключить, что приводит к излишнему расходу электроэнергии.

Задача обеспечить включение света на заданный интервал времени, например, у входной двери в квартиру, по следующему алгоритму:

- 1) перед входной дверью установлен датчик освещения (F1) и кнопка включения света ТАЙМЕР (SB1);
- 2) при кратковременном нажатии на кнопку ТАЙМЕР, при недостаточном естественном освещении, светильник должен включаться на интервал времени 1 мин – этого времени достаточно, чтобы найти замочную скважину и ключом открыть дверь;
- 3) при удерживании нажатой кнопки ТАЙМЕР в течение 2 с светильник должен включаться на интервал времени 3 мин независимо от внешнего освещения – этот режим может потребоваться при уборке коридора;
- 4) предусмотреть возможность управления работой светильника по командам от внешних управляющих устройств или при помощи выключателя СВЕТ (SA1), независимо от внешнего освещения. Этот режим удобен во время приема гостей или для дальнейшей автоматизации квартиры в рамках программы «умный дом»;
- 5) предусмотреть возможность включения светильника только в вечернее и ночное время.

#### Выбор программируемого прибора

Для реализации данной задачи управляющий автомат должен иметь встроенные часы реального времени, три входа и один выход, что обеспечивают приборы из серии ПР110, имеющие последнюю букву «Ч» в обозначении.

При выполнении автоматики на основе ПР110-24.8Д4Р-Ч можно воспользоваться схемой, приведенной на рисунке 4.1 (номера входов и выходов для ПР выбраны произвольно).

#### Создание коммутационной программы

На этапе планирования составляется схема реализации поставленной задачи на основе логических элементов и функциональных блоков, доступных в программе (см. Приложения А и Б).

Схема может иметь вид, приведенный на рисунке 4.2.

Выход Q2 схемы используется как контрольный для проверки функционирования логической части (элементов D1–D6). Выход Q1 является основным и может включиться только в заданные таймером D7 интервалы времени, при соблюдении логических условий, обеспечиваемых элементами D1–D6.

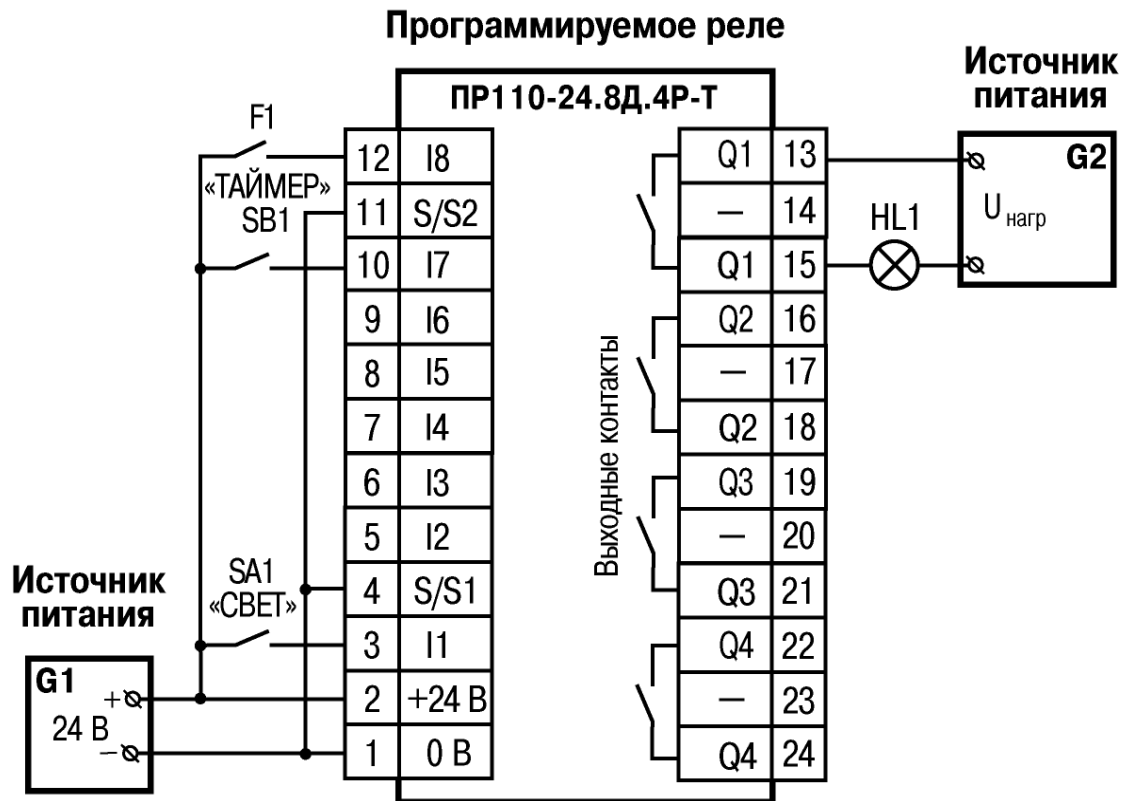


Рисунок 4.1 – Электрическая схема подключения ПР, SA1 – переключатель OFF-ON; SB1 – кнопка без фиксации OFF-ON; HL1 – световая лампа

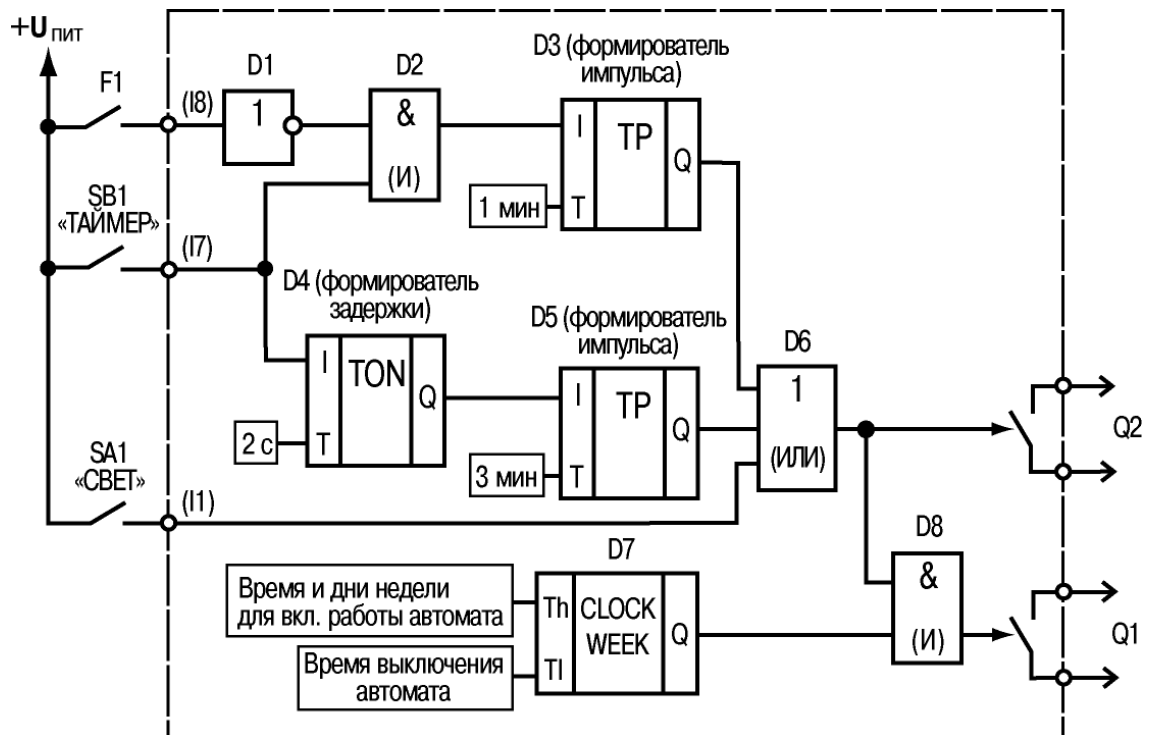


Рисунок 4.2 – Схема электрическая принципиальная для управляющей программы «Свет\_Logic»

#### Описание работы схемы:

1) при кратковременном (менее 2 с) нажатии на кнопку ТАЙМЕР (SB1), лог. «1» поступает на вход логического элемента И (D2). Так как на втором входе D2 также присутствует лог. «1», на выходе D2 появится лог. «1», которая запустит

формирователь импульса (D3) длительностью 1 мин. Этот импульс через элемент ИЛИ (D6) поступит на выход Q2;

2) контакты датчика F1 замыкают цепь только при хорошем естественном освещении – в этом случае лог. «1» на входе D1 преобразуется в лог. «0» на выходе и поступает на вход логического элемента И (D2), что блокирует работу D2. При этом нажатие кнопки ТАЙМЕР не включит выход Q2;

3) при удержании нажатой кнопки ТАЙМЕР более 2 с, на выходе блока D4 появится лог. «1», которая запустит формирователь импульса (D5) длительностью 3 мин. Этот импульс через элемент ИЛИ (D6) поступит на выход Q2 (интервал действия импульса не зависит от состояния датчика F1);

4) при включении тумблера «СВЕТ» (SA1) лог. «1» через элемент ИЛИ (D6) поступит на выход Q2 (на время включения SA1);

5) сигнал лог. «1» с выхода Q2 поступает на вход логического элемента И (D8), а если на второй вход этого элемента также поступает лог. «1» (от таймера реального времени D7) – на выходе Q1 появится лог. «1» (выходное реле Q1 включится).

Программа, обеспечивающая работу ПР, будет иметь вид, приведенный на рисунке 4.3.

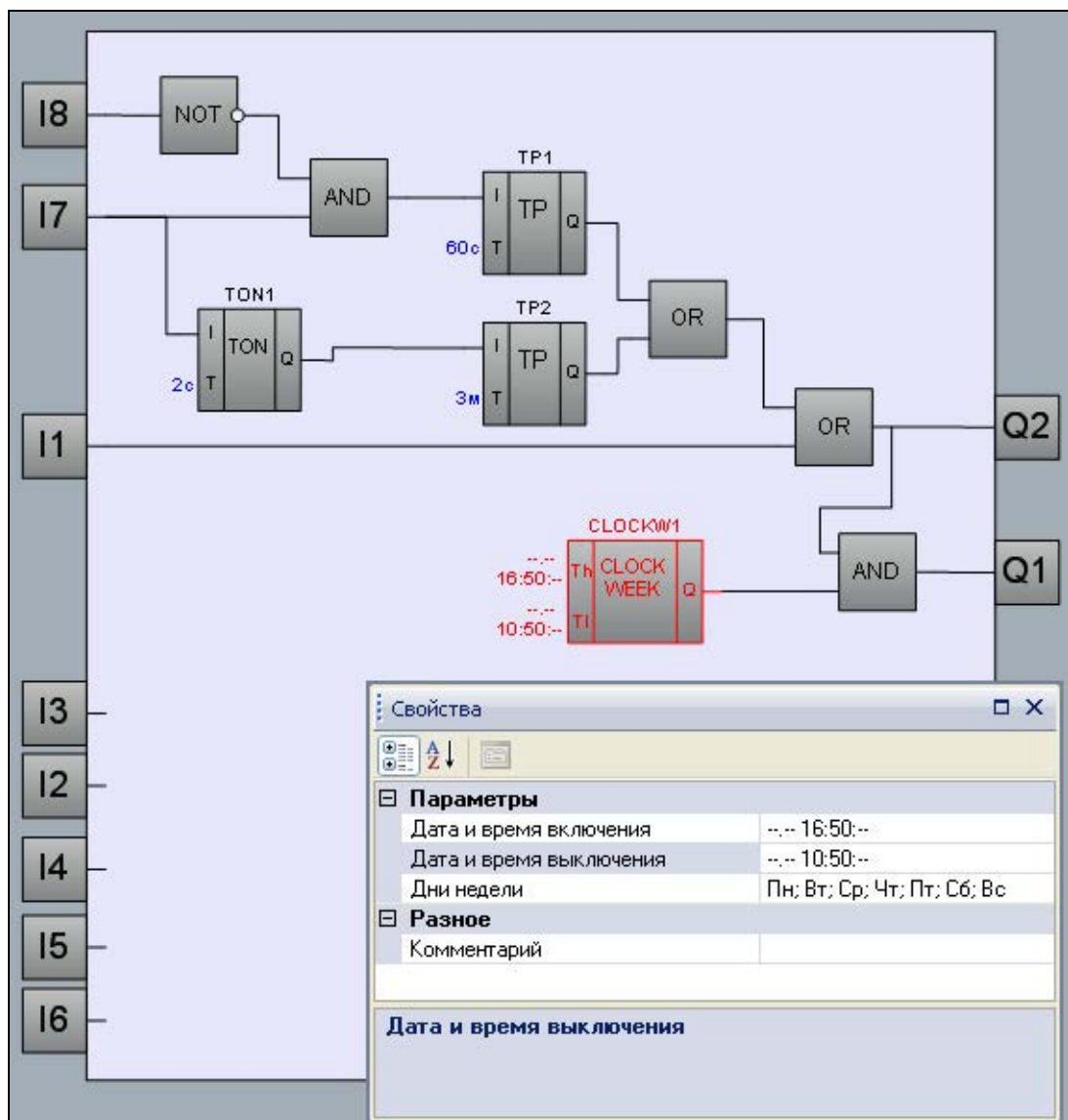


Рисунок 4.3 – Управляющая программа «Свет\_Logic» на основе ФБ программы OWEN Logic (с открытой вкладкой свойств для блока CLOCKW1)

## 4.2 Автоматическое управление электродвигателем мешалки

При производстве пищевых продуктов часто требуется перемешивать компоненты (например, молоко или сливки на молочной ферме) в течение определенного времени.

Задача обеспечить работу установки по следующему алгоритму:

- 1) необходимы два режима работы: «Автоматический» и «Ручной», переключаемых тумблером «РЕЖИМ» (SA1);
- 2) в режиме «Автоматический», при включении оператором установки кнопкой ПУСК (SB1), производится автоматическое включение и отключение электродвигателя через заданные интервалы времени (15 с – включен, 10 с – отключен). Отключение установки производится через интервал 5 мин или оператором при помощи кнопки СТОП (SB2);
- 3) в режиме «Ручной» производится прямое управление работой электродвигателя от кнопок ПУСК и СТОП (без временных интервалов отключения);
- 4) при перегрузке электродвигателя (на котором устанавливается соответствующий датчик – F1) должно срабатывать автоматическое отключение установки с индикацией режима «Неисправность» лампой (HL1) и звуковым прерывистым сигналом (интервал повторения звукового сигнала 3 с);
- 5) звуковой сигнал должен отключаться кнопкой СБРОС (SB3);
- 6) с помощью кнопки КОНТРОЛЬ (SB4) проверяется исправность элементов сигнализации – работа лампы и звукового сигнала.

### Выбор программируемого прибора

Для реализации данной задачи управляющий прибор должен иметь шесть входов (по числу управляющих сигналов) и три выхода, что обеспечивает любая модель прибора из серии ПР110.

При выполнении автоматики на основе ПР110-24.8Д4Р можно воспользоваться схемой, приведенной на рисунке 4.4.

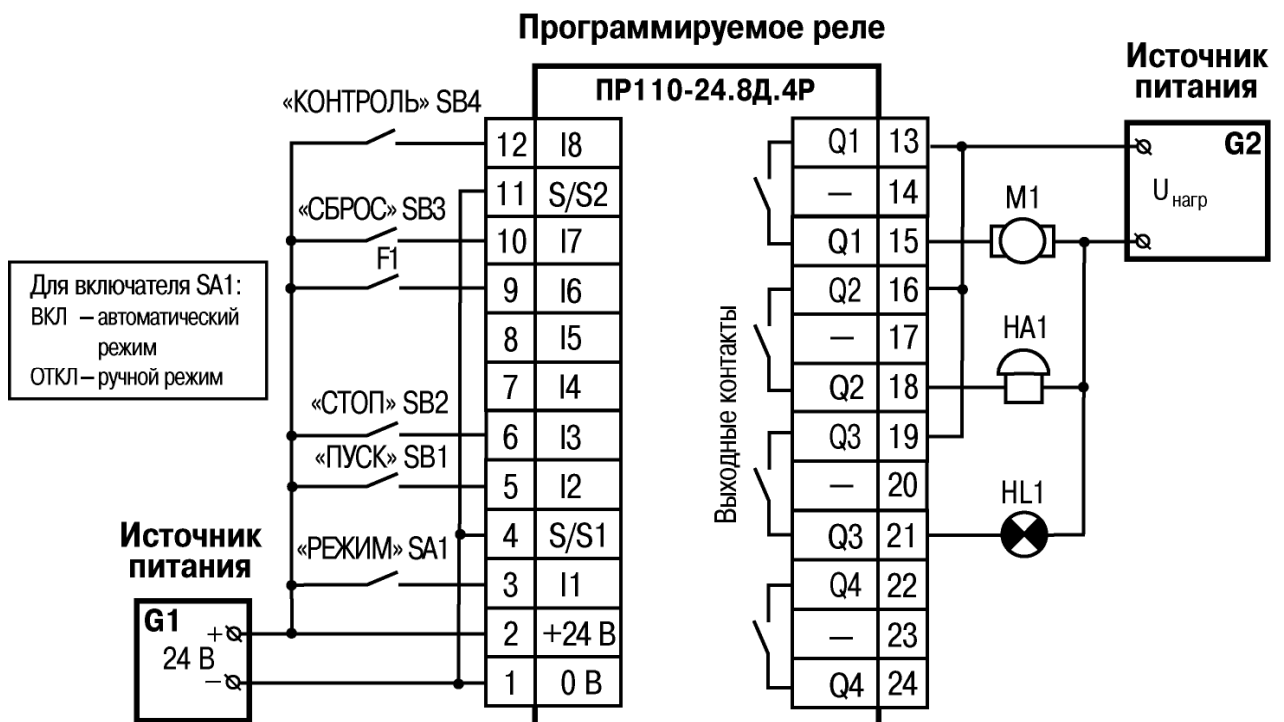


Рисунок 4.4 – Электрическая схема подключения ПР, SA1 – переключатель OFF-ON; SB1–SB4 – кнопки без фиксации OFF-ON; M1 – электродвигатель; HA1 – звонок; HL1 – индикаторная лампа

Работу программы прибора можно представить в виде цепей схемы, приведенной на рисунке 4.5 (выходы Q1–Q3 соответствуют двум контактам клеммника ПР).

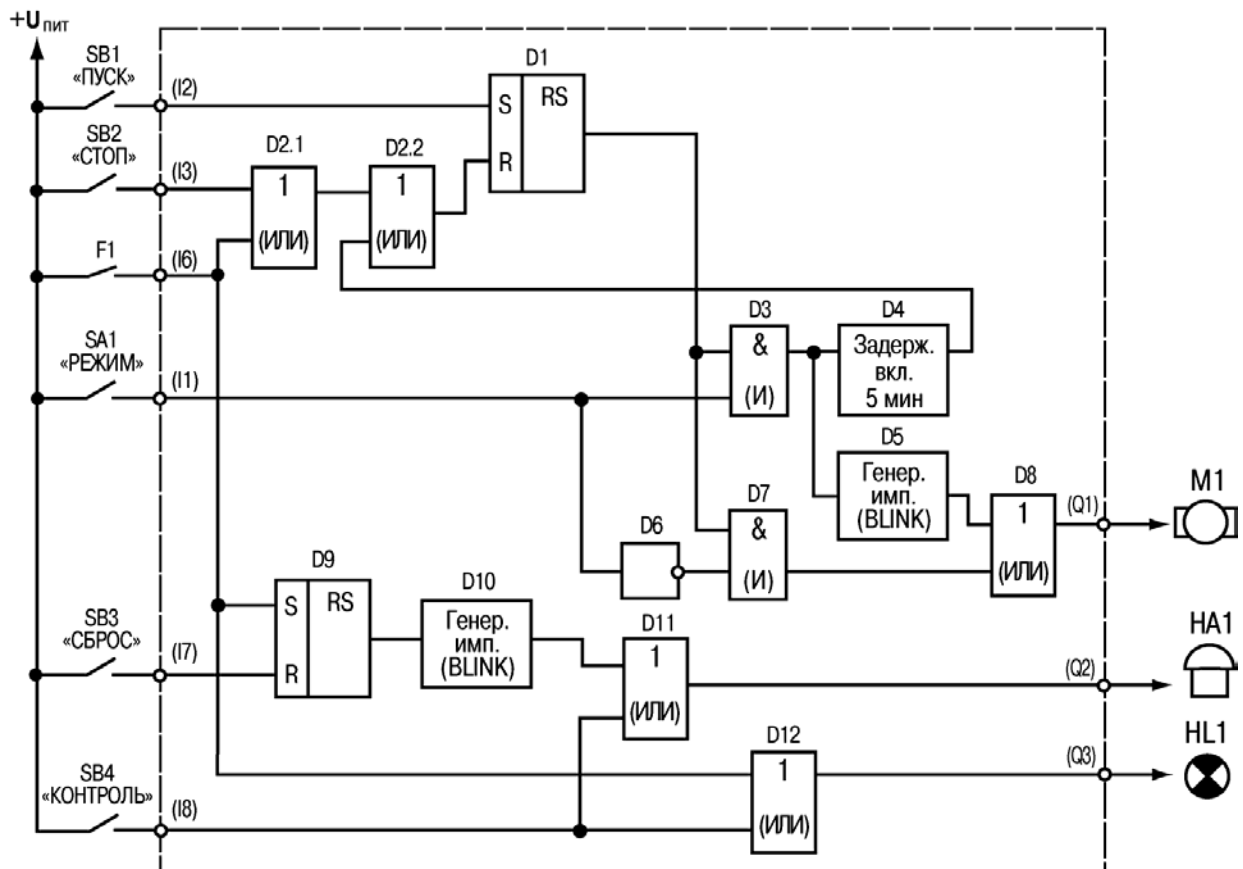


Рисунок 4.5 – Схема электрическая принципиальная для управляющей программы «Мешалка\_Logig»

#### Описание работы схемы:

**Цепь входа I2 (включение установки):** при нажатии кнопки ПУСК (SB1) на входе S (D1) появляется лог. «1» – на выходе RS-триггера D1 установится лог. «1». Этот сигнал поступает дальше, в зависимости от состояния выключателя SA1:

– при разомкнутых контактах SA1 (РЕЖИМ – Ручной), лог. «1» проходит через элементы D8, D9 и поступает на выход Q1 (контактами выходного реле электромотор M1 включится);

– при замкнутых контактах SA1 (РЕЖИМ – Автоматический), лог. «1» проходит только через элемент D3 для запуска работы блоков D4, D5.

**Цепь входа I3 (отключение установки):** при нажатии кнопки СТОП (SB2) или срабатывании датчика F1 на входе R (D1) появляется лог. «1» – на выходе RS-триггера (D1) установится лог. «0» (включение выхода Q1 блокируется).

**Цепь входа I1 (формирование интервалов работы электромотора):** при включенном триггере D1 и замкнутых контактах выключателя SA1 (РЕЖИМ – Автоматический):

– сигнал лог. «1» от SA1 проходит через элемент D3 и поступает на D4 (формирователь импульса с задержкой 5 мин). Этот импульс, проходя через

элемент ИЛИ (D2), поступит на вход R триггера D1 – установка отключится.

Сигнал лог. «1» с выхода D3 поступает на D5 – генератор импульсов с параметрами: лог. «1» – 15 с, лог. «0» – 10 с. Эти импульсы с выхода генератора проходят через элемент ИЛИ (D8) и поступают на выход Q1 для управления работой электромотора M1.

При разомкнутых контактах выключателя SA1 (РЕЖИМ – Ручной) логический элемент D3 заблокирован и сигнал с выхода триггера D1 через элемент D6 сразу поступает на выход Q1, т. е. в этом режиме состояние выхода триггера непосредственно управляет работой электромотора M1.

**Цепь входа I6 (включение звуковой сигнализации):** при срабатывании датчика F1 на входе S (D9) появляется лог. «1» – на выходе RS-триггера D9 установится лог. «1», которая включает генератор D10 с параметрами: лог. «1» – 3 с, лог. «0» – 3 с. Эти импульсы проходят через элемент ИЛИ (D12) и поступают на выход Q2 для управления работой звонка HA1.

**Цепь входа I7 (отключение звуковой сигнализации):** при нажатии кнопки СБРОС (SB3) на входе R (D9) появляется лог. «1» – на выходе RS-триггера установится лог. «0» – генератор D10 отключится.

**Цепь входа I8 (контроль работы лампы и звуковой сигнализации):** при нажатой кнопке КОНТРОЛЬ (SB4) лог. «1» через элемент ИЛИ (D11) поступает на включение выхода Q3 – лампа HL1 будет непрерывно светиться.

Выход Q3 включится также, если контакты датчика F1 замкнуты.

При нажатой кнопке КОНТРОЛЬ (SB4) лог. «1» через элемент ИЛИ (D12) поступает на включение выхода Q2 – звонок HA1 будет непрерывно работать.

Коммутационная программа, обеспечивающая работу ПР, будет иметь вид, приведенный на рисунке 4.6.

### **Примечания**

1. Оставшиеся свободные два входа и один выход, при необходимости, можно использовать для введения дополнительных функций. Например, переключать четыре разных интервала длительности автоматической работы электромотора или же изменять другие рабочие параметры установки.

2. Технологический цикл работы установки можно сделать полностью автоматическим, если схему дополнить инкрементным счетчиком (СТ), выходным сигналом которого можно выключать триггер D1.



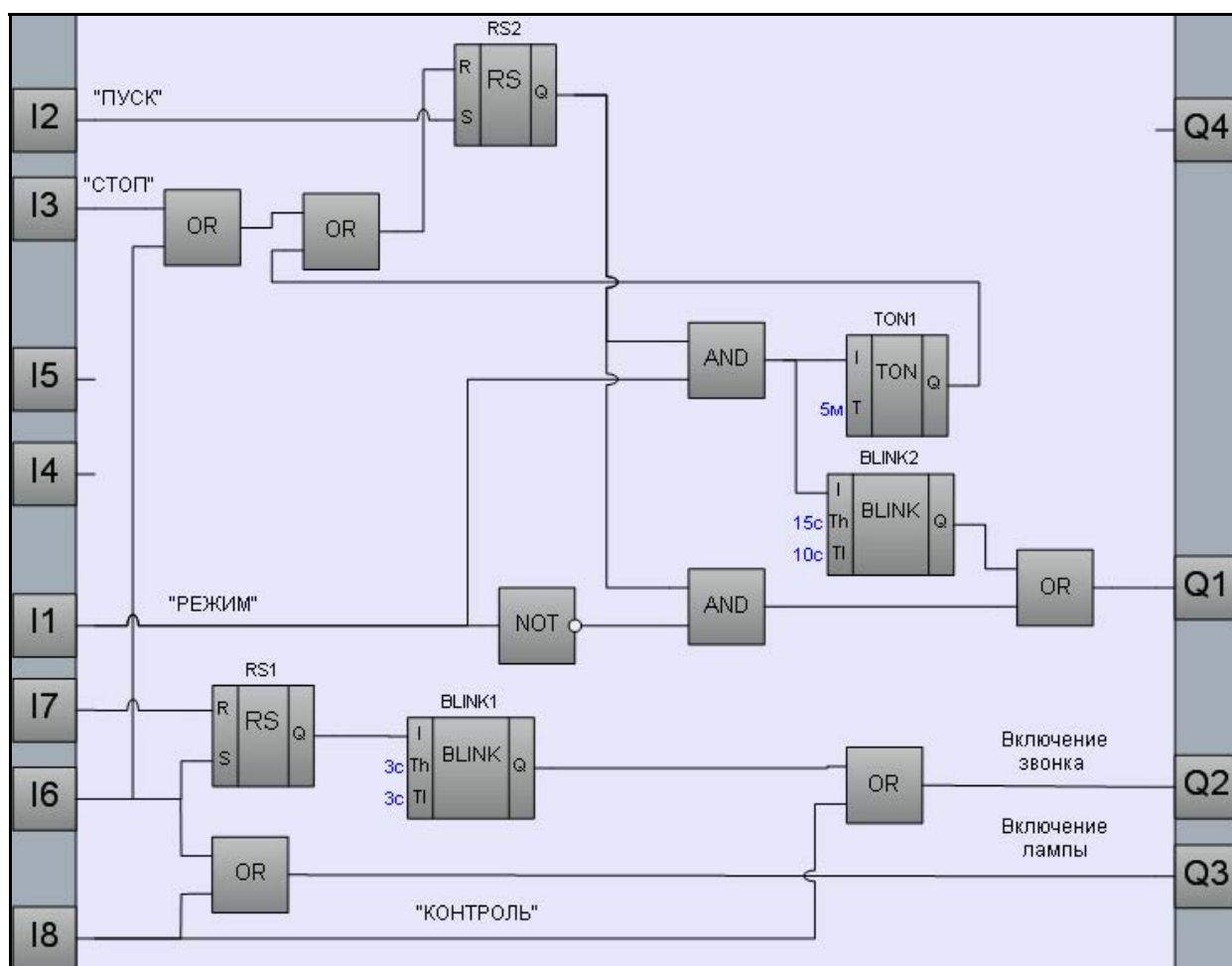


Рисунок 4.6 – Управляющая программа «Мешалка\_Logis» на основе ФБ среды программирования OWEN Logic

## Приложение А. Функции логических элементов программы

### А.1 Битовые операции

При использовании функций **И** и **ИЛИ** следует учитывать, что не подключенные входы логических элементов к входам ПР или другим элементам в программе, будут иметь следующие состояния:

- для элемента **И (AND)** – логическая «1»;
- для элемента **ИЛИ (OR)** – логический «0».

В этом случае логические элементы выполняют функцию повторителя сигнала.

Для увеличения числа входов у логических элементов используется их каскадное включение, например, как это показано на рисунке А.1.

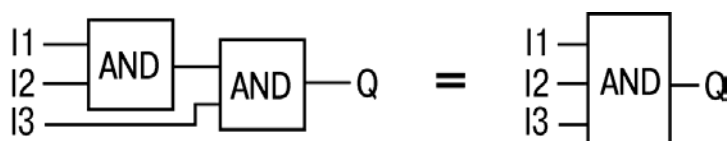


Рисунок А.1 – Пример каскадного включения логических элементов И

### Функция «НЕ» (NOT)



I1	Q
0	1
1	0

Элемент используется для инвертирования сигнала.

На выходе элемента логическая «1» (выход включен), если на входе логический «0» (контакты разомкнуты) и наоборот – инвертируется сигнал.

Если на входы функции заведены целочисленные значения, то операция будет произведена над каждым битом в отдельности.

### Функция «И» (AND)



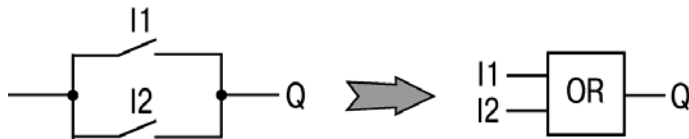
I1	I2	Q
0	0	0
0	1	0
1	0	0
1	1	1

Элемент используется для выполнения логических операций. На выходе элемента логическая «1» (выход включен), если на всех входах логическая «1» (все входы включены – контакты замкнуты).

Работе соответствует приведенная таблица состояний.

Если на входы функции заведены целочисленные значения, то операция будет произведена над каждым битом в отдельности.

## Функция «ИЛИ» (OR)



I1	I2	Q
0	0	0
0	1	1
1	0	1
1	1	1

Элемент используется для выполнения логических операций. На выходе элемента логическая «1» (выход включен), если хотя бы на одном входе логическая «1» (контакты замкнуты).

Работе соответствует приведенная таблица состояний.

Если на входы функции заведены целочисленные значения, то операция будет произведена над каждым битом в отдельности.

## Функция «исключающее ИЛИ» (XOR)



I1	I2	Q
0	0	0
0	1	1
1	0	1
1	1	0

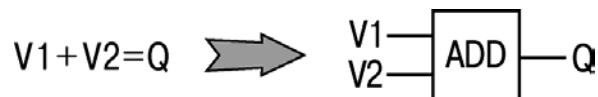
Элемент используется для выполнения логических операций. На выходе элемента логическая «1» (выход включен), только если на любом из входов логическая «1».

Работе соответствует приведенная таблица состояний.

Если на входы функции заведены целочисленные значения, то операция будет произведена над каждым битом в отдельности.

## А.2 Арифметические операции

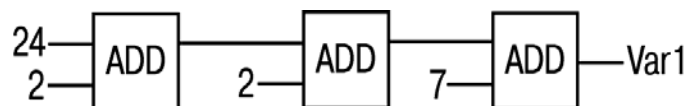
### Сложение (ADD)



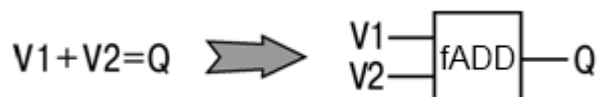
Результатом выполнения операции на выходе является сумма входных значений.

Если при выполнении операции значение числа получается больше 4294967295 (32 бита), выходящие за разрядность 32 бита биты отсекаются.

Пример применения элементов для сложения чисел  $24 + 2 + 4 + 7 = \text{Var1}$ :

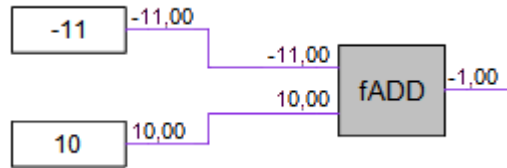


### Сложение чисел с плавающей точкой (fADD)



Результатом выполнения операции на выходе является сумма входных значений.

Демонстрация работы элемента в режиме симулятора:



## Вычитание (SUB)

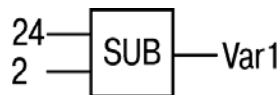


Результатом выполнения операции на выходе является разность входных значений.

Если при выполнении операции на первом входе (V1) значение числа меньше второго (V2), результатом будет число, полученное сложением младшего числа плюс 0x100000000 (4294967296) минус значение числа большего:

$$(V1 + 0x100000000) - V2 = Q.$$

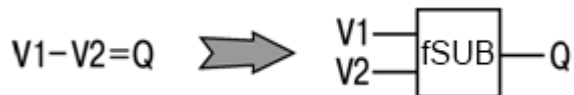
Пример применения элемента для вычитания чисел  $24 - 2 = \text{Var1} = 22$ :



Пример применения элемента для вычитания чисел, где V1 меньше V2:

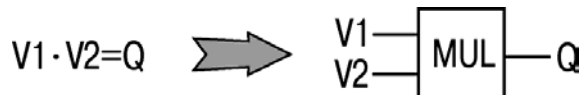
$$2 - 24 = \text{Var1} = 4294967274.$$

## Вычитание чисел с плавающей точкой (fSUB)



Результатом выполнения операции на выходе является разность входных значений.

## Умножение (MUL)

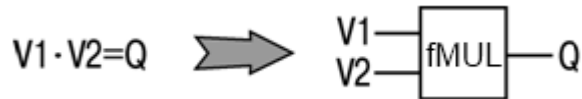


Результатом выполнения операции на выходе является перемножение входных значений.

Если при выполнении операции значение числа получается больше 4294967295 (32 бита), выходящие за разрядность 32 бита биты отсекаются.

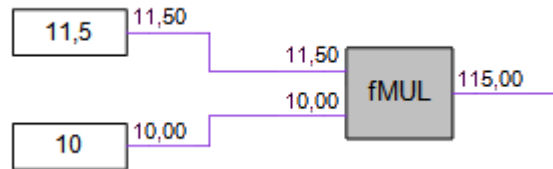
Пример применения элементов для перемножения чисел  $24 \cdot 2 \cdot 4 \cdot 7 = \text{Var1}$ :



**Умножение чисел с плавающей точкой (fMUL)**

Результатом выполнения операции на выходе является перемножение входных значений.

Демонстрация работы элемента в режиме симулятора:

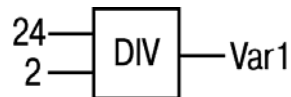
**Деление (DIV)**

Результатом выполнения операции на выходе является деление входных значений.

Если в результате деления получаются доли целого числа, то на выходе производится округление значения до целого числа в меньшую сторону.

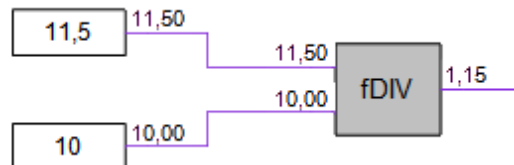
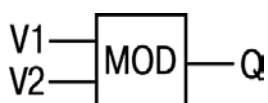
При делении на 0 на выходе элемента будет 0xFFFFFFFF.

Пример применения элемента для деления чисел  $24/2 = \text{Var1}$ :

**Деление чисел с плавающей точкой (fDIV)**

Результатом выполнения операции на выходе является деление входных значений.

Демонстрация работы элемента в режиме симулятора:

**Деление с остатком (MOD)**

Результатом выполнения операции на выходе является остаток от деления входных значений.

Пример применения элемента для выделения целого остатка от деления чисел 22 на 3 (Var1 = 1):

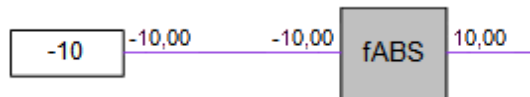


### Абсолютное значение числа с плавающей точкой (fABS)



Результатом выполнения операции на выходе является деление входных значений.

Демонстрация работы элемента в режиме симулятора:



### Возведение в степень числа с плавающей точкой (fPOW)

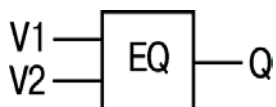


Результатом выполнения операции на выходе является деление входных значений.

Демонстрация работы элемента в режиме симулятора:

## А.3 Операции сравнения и выбора

### Равно (EQ)

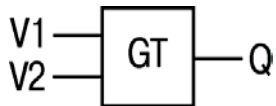


V1 = V2; Q = 1  
V1 > V2; Q = 0  
V1 < V2; Q = 0

Результатом выполнения операции на выходе является лог. «1», если входные числа равны.

Примеры применения элементов:

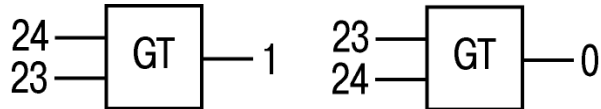
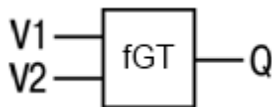


**Больше (GT)**

$V1 = V2; Q = 0$   
 $V1 > V2; Q = 1$   
 $V1 < V2; Q = 0$

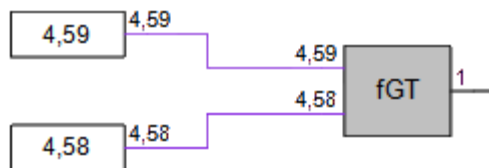
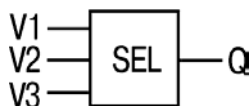
Результатом выполнения операции на выходе является лог. «1», если входное число на верхнем входе (V1) больше числа на втором входе (V2).

Примеры применения элементов:

**Больше, числа с плавающей точкой (fGT)**

$V1 = V2; Q = 0$   
 $V1 > V2; Q = 1$   
 $V1 < V2; Q = 0$

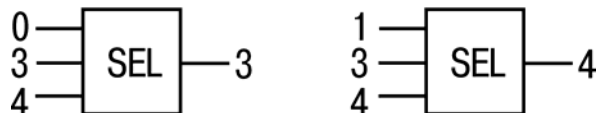
Результатом выполнения операции на выходе является лог. «1», если входное число на верхнем входе (V1) больше числа на втором входе (V2).

**Выбор (SEL)**

$V1 = 0; Q = V2$   
 $V1 = 1; Q = V3$

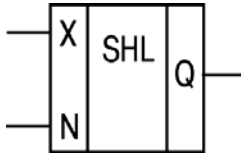
Результатом выполнения операции на выходе является входной сигнал V2, если V1 равен лог «0», и V3, если V1 равен лог «1».

Примеры применения элементов:



## А.4 Операции логического битового сдвига и преобразования

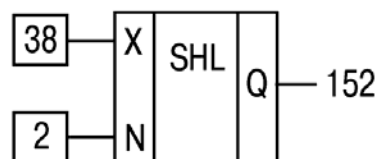
### Побитный логический сдвиг влево (SHL)



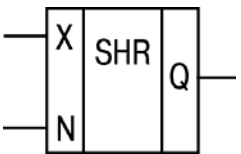
Элемент используется для выполнения операции побитного логического сдвига операнда X влево на N бит с дополнением нулями справа.

Пример применения для сдвига числа 38(десятичное) = 00100110(двоичное):

00100110 → 10011000



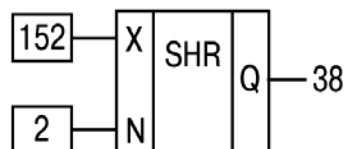
### Побитный логический сдвиг вправо (SHR)



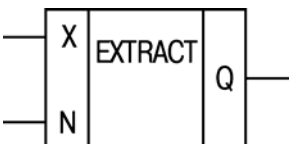
Элемент используется для выполнения операции побитного логического сдвига операнда X вправо на N бит с дополнением нулями слева.

Пример применения для сдвига числа 152(десятичное) = 10011000(двоичное):

10011000 → 00100110



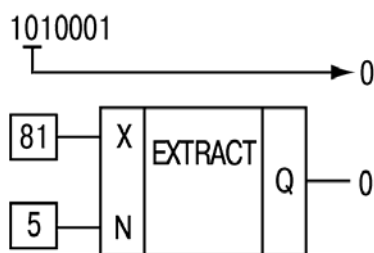
### Чтение бита (EXTRACT)



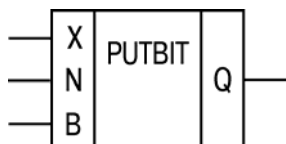
Элемент используется для выполнения операции чтения значения бита N в числе на входе X. Биты нумеруются с нуля. Число на входе X задается в десятичной системе. Выходное значение всегда двоичное (0 или 1).

Пример применения для чтения пятого бита из числа 81(десятичное) = 1010001(двоичное):



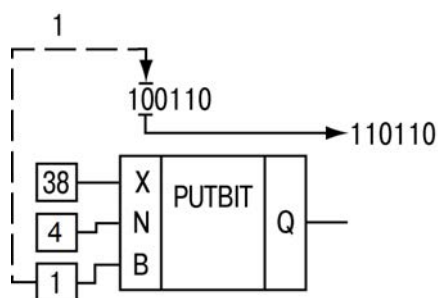


### Запись бита (PUTBIT)



Элемент используется для выполнения операции записи в числе X значения бита N в состояние, указанное на входе B (лог. «0» или «1»). Число на входе X задается в десятичной системе. Выходное значение всегда двоичное (0 или 1).

Пример применения записи четвертого бита в лог. «1» для числа 38(десятичное) = 100110(двоичное):



### Дешифратор (DC32)



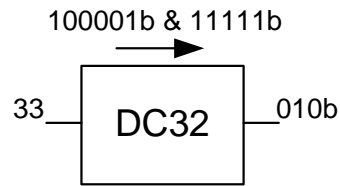
Элемент используется для выполнения операции преобразования двоичного кода на входе в позиционный код на выходе.

Перед выполнением операции над значением на входе предварительно выполняется побитовая логическая операция «И» с операндом 0x1F (11111b).

Работе элемента соответствует приведенная таблица состояний.

Двоичный код					Позиционный код									
5	4	3	2	1	32	31	· · ·	6	5	4	3	2	1	
0	0	0	0	0	0	0		0	0	0	0	0	1	
0	0	0	0	1	0	0		0	0	0	0	1	0	
0	0	0	1	0	0	0		0	0	0	1	0	0	
0	0	0	1	1	0	0		0	0	1	0	0	0	
		1	0	0	0	0			1					
...					...			...						
1	1	1	0	1	0	0		0	0	0	0	0	0	
1	1	1	1	0	0	1		0	0	0	0	0	0	
1	1	1	1	1	1	0		0	0	0	0	0	0	

Пример применения:



### Шифратор (CD32)



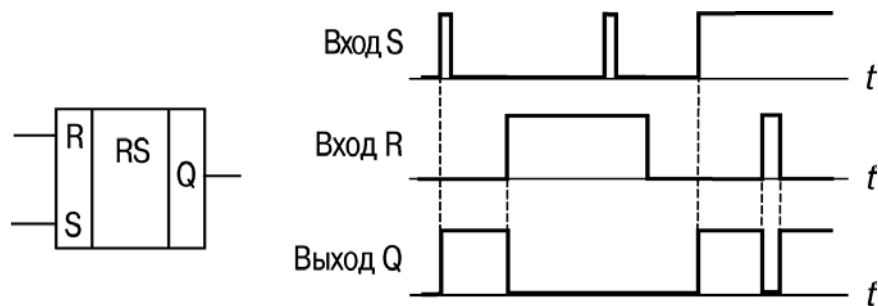
Элемент используется для выполнения операции преобразования позиционного кода на входе в двоичный код на выходе.

В случае, если входное двоичное значение имеет более одной логической «1» в разрядах, работа элемента ведется только со старшим единичным разрядом.

Работе элемента соответствует приведенная таблица состояний в пункте «Дешифратор».

## Приложение Б. Функциональные блоки программы

### RS-триггер с приоритетом выключения (RS)

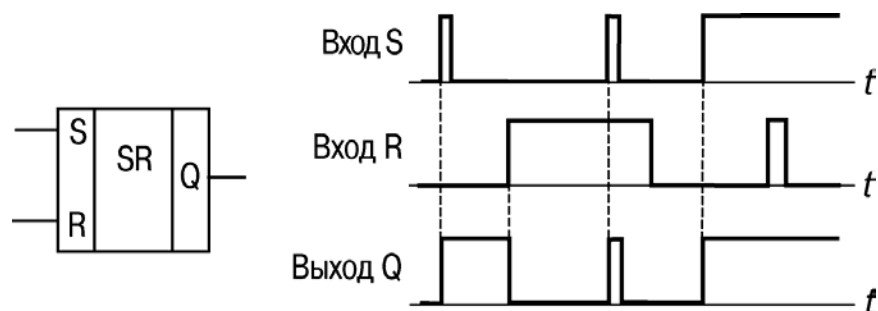


Блок используется для переключения с фиксацией состояния при поступлении коротких импульсов на соответствующий вход. Работу поясняет приведенная на рисунке диаграмма.

На выходе блока появится логическая «1» по фронту сигнала на входе S.

При одновременном поступлении сигналов на входы приоритетным является сигнал входа R.

### SR-триггер с приоритетом включения (SR)

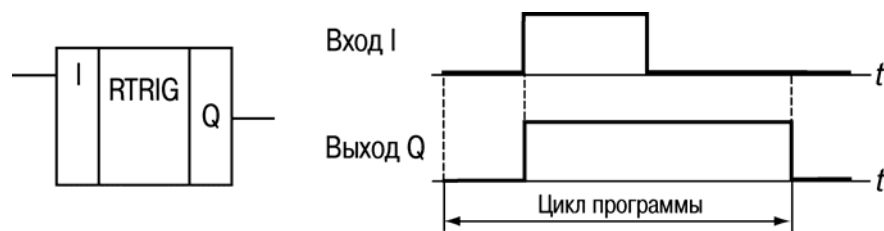


Блок используется для переключения с фиксацией состояния при поступлении коротких импульсов на соответствующий вход. Работу поясняет приведенная на рисунке диаграмма.

На выходе блока появится логическая «1» по фронту сигнала на входе S.

При одновременном поступлении сигналов на входы приоритетным является сигнал входа S.

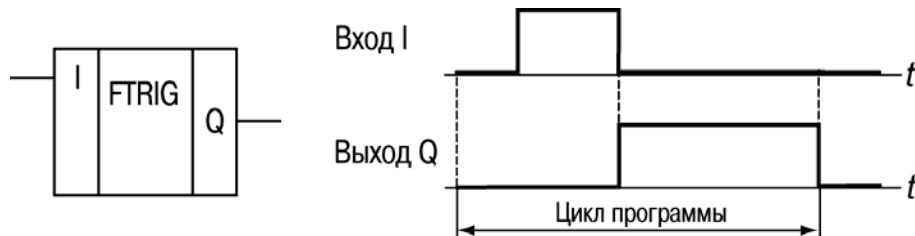
## Детектор переднего фронта импульса (RTRIG)



Блок используется в случае необходимости иметь реакцию на изменение состояния дискретного входного сигнала. На выходе генерируется единичный импульс по переднему фронту входного сигнала.

Работу поясняет приведенная на рисунке диаграмма.

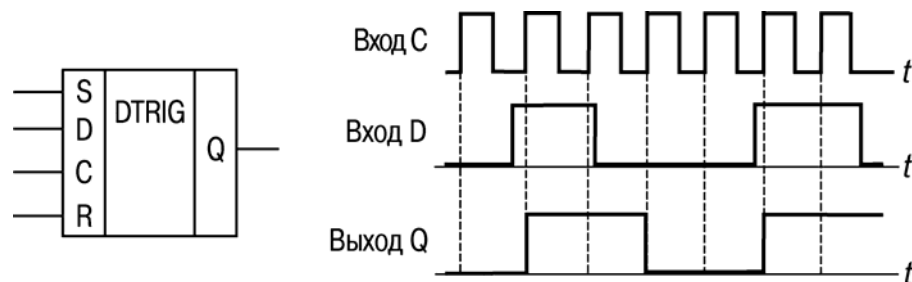
## Детектор заднего фронта импульса (FTRIG)



Блок используется в случае необходимости иметь реакцию на изменение состояния дискретного входного сигнала. На выходе генерируется единичный импульс по заднему фронту входного сигнала.

Работу поясняет приведенная на рисунке диаграмма.

## D-триггер (DTRIG)



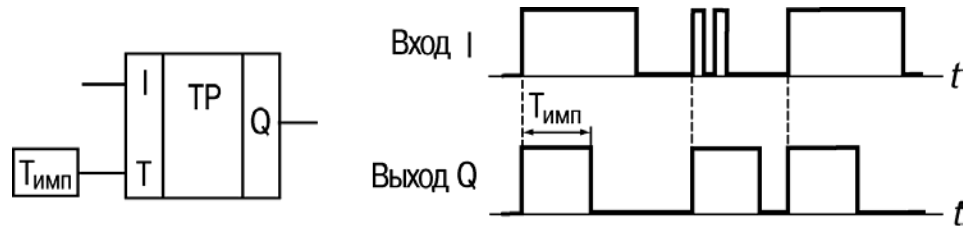
Блок используется для формирования импульса включения выхода на интервал времени импульса на входе D, при этом выходной интервал будет синхронизирован с тактовой частотой на входе C.

На выходе Q блока появится логическая «1» по фронту тактовых импульсов на входе C при наличии логической «1» на входе D. Возврат выхода в «0» произойдет по фронту тактовых импульсов на входе C при логическом «0» на входе D.

Работу поясняет приведенная на рисунке диаграмма.

Вход S обеспечивает принудительную установку выхода Q в состояние логической единицы; вход R является приоритетным и обеспечивает установку выхода Q в состояние логического «0».

## Импульс включения заданной длительности (TP)

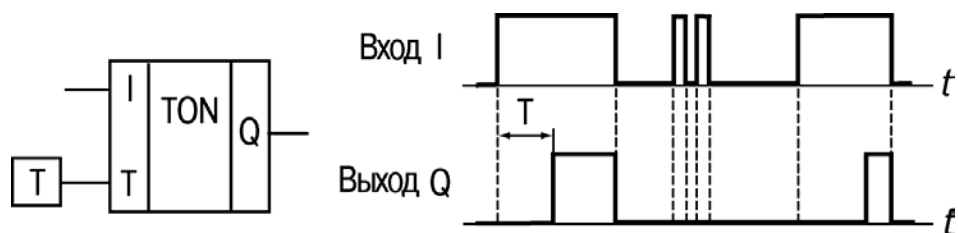


Блок используется для формирования импульса включения выхода на заданный интервал времени. Работу поясняет приведенная на рисунке диаграмма.

На выходе Q блока появится логическая «1» по фронту входного сигнала (I). После запуска выход Q не реагирует на изменение значения входного сигнала на интервале Тимп, а по истечении этого интервала сбрасывается в «0».

Допустимый диапазон значений Тимп=Т – от 0 до 4147200000 мс, или 48 дней.

## Таймер с задержкой включения (TON)

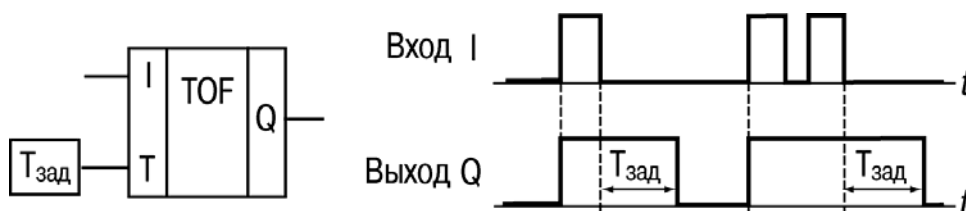


Блок используется для операции задержки передачи сигнала. Работу поясняет приведенная на рисунке диаграмма.

На выходе Q блока появится логическая «1» с задержкой относительно фронта входного сигнала (I). Выход включается логической «1» на входе продолжительностью не менее длительности Т, а выключается по спаду входного сигнала.

Допустимый диапазон значений Т – от 0 до 4147200000 мс, или 48 дней.

## Таймер с задержкой отключения (TOF)

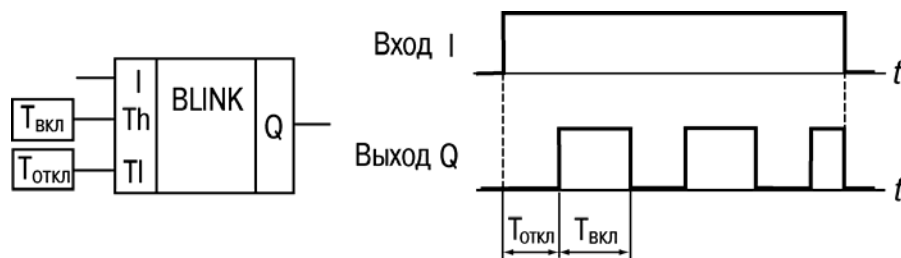


Блок используется для задержки отключения выхода. Работу поясняет приведенная на рисунке диаграмма.

На выходе блока появится логическая «1» по фронту сигнала на входе I, а начало отсчета времени задержки отключения (Тзад) происходит по каждому спаду входного сигнала. После отключения входного сигнала на выходе появится логический «0» с задержкой Тзад.

Допустимый диапазон значений Тзад – от 0 до 4147200000 мс, или 48 дней.

## Генератор прямоугольных импульсов (BLINK)

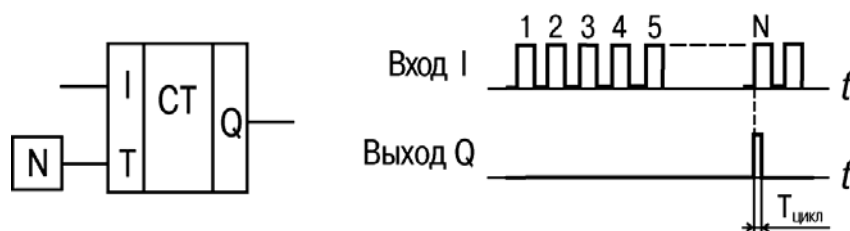


Блок используется для формирования прямоугольных импульсов пульсации.

На выходе Q генератора формируются импульсы с заданными параметрами длительности включенного ( $T_{вкл}$  – логическая «1») и отключенного ( $T_{откл}$  – логический «0») состояния на время действия управляющего сигнала на входе I (логической «1»). Работу поясняет приведенная на рисунке диаграмма.

Допустимый диапазон значений  $T_{вкл}$  и  $T_{откл}$  – от 0 до 4233600000 мс, или 49 дней.

## Инкрементный счетчик с автосбросом (СТ)

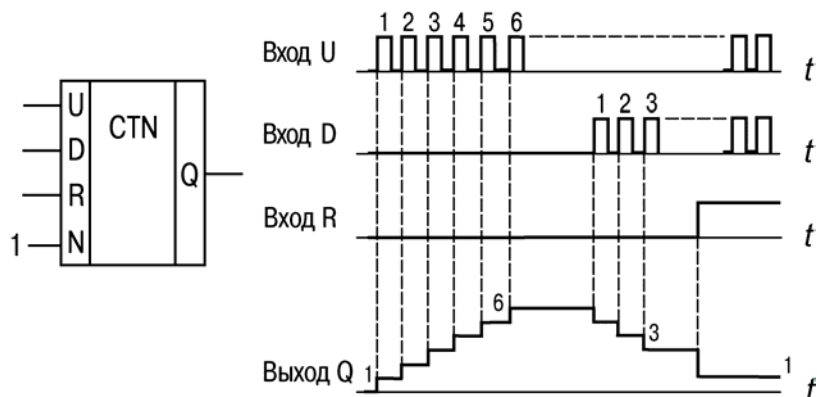


Блок используется для подсчета заданного числа импульсов N (вход T – уставка числа импульсов). Работу поясняет приведенная на рисунке диаграмма.

На выходе Q счетчика появится импульс логической «1» с длительностью рабочего цикла прибора ( $T_{цикл}$ ), если число приходящих на вход I импульсов достигнет установленного значения N (на входе T).

Допустимый диапазон значений числа импульсов N – от 0 до 65535.

## Универсальный счетчик (CTN)



Блок является счетчиком, который используется для прямого и обратного счета. Операция «прямой счет» выполняется по переднему фронту импульса на входе прямого счета (U), что увеличивает значение выходного сигнала «Q». Импульсы, приходящие на вход D («обратный счет»), уменьшают значение

выхода «Q».

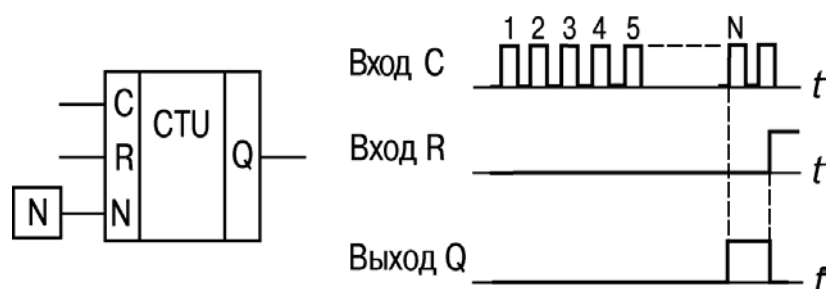
При поступлении на вход R логической «1», выход счетчика «Q» устанавливается в значение входа N.

Работу блока поясняет приведенная на рисунке диаграмма.

При одновременном поступлении сигналов на входы U и D приоритетным является сигнал входа U.

Допустимый диапазон значений числа импульсов N – от 0 до 65535.

### Инкрементный счетчик (CTU)

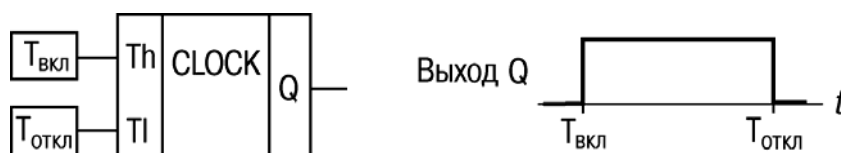


Блок используется для подсчета числа импульсов, приходящих на вход C. Работу поясняет приведенная на рисунке диаграмма – на выходе Q счетчика появится импульс лог. «1», если число приходящих на вход импульсов достигнет установленного значения на входе N (N – уставка).

Допустимый диапазон значений числа импульсов N – от 0 до 65535.

Счетчик сбрасывается в 0 по переднему фронту импульса на входе R. При одновременном поступлении сигналов на входы приоритетным является сигнал входа R.

### Интервальный таймер (CLOCK)



Блок используется для формирования импульса включения выхода по часам реального времени. Работу поясняет приведенная на рисунке диаграмма.

Время включения (Твкл) и отключения (Тоткл) выходов устанавливают в качестве параметров блока.

Допустимый диапазон значений Твкл и Тоткл – от 0,00 с до 24 ч.

В случае если значение время отключения задано раньше времени включения (рисунок Б.1), диаграмма переключений будет иметь вид, приведенный на рисунке Б.2 (выход будет включен до момента времени отключения).

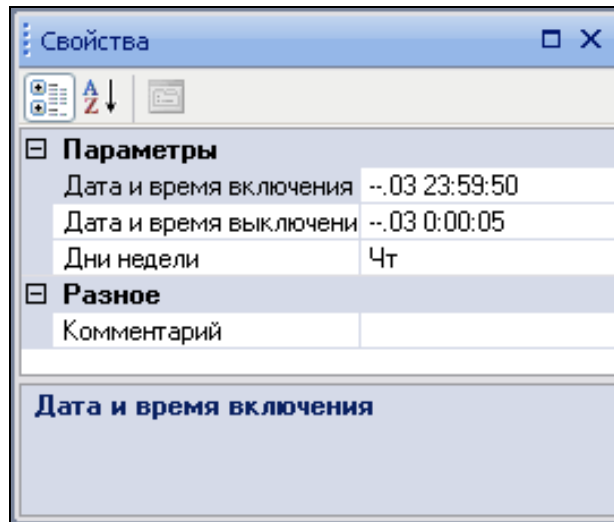


Рисунок Б.1 – Вкладка «Свойства» для настройки временных параметров работы блока

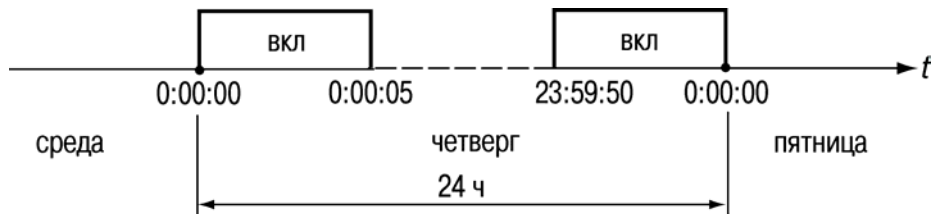
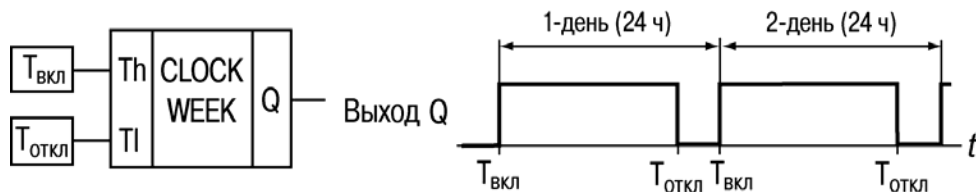
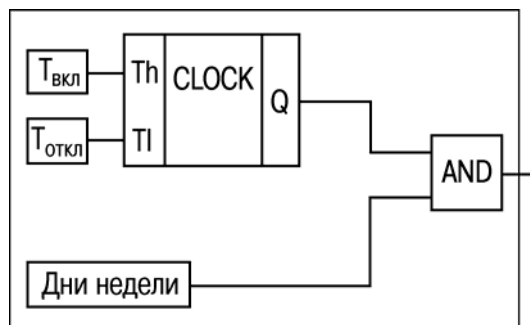


Рисунок Б.2 – Диаграмма включения выхода блока

### Интервальный таймер с недельным циклом (CLOCK WEEK)



Блок используется для формирования импульса включения выхода по часам реального времени, с учетом дней недели. Работу поясняет приведенная на рисунке диаграмма. Внутренняя структура блока имеет вид:



Время включения ( $T_{вкл}$ ), отключения ( $T_{откл}$ ) выхода и дни недели работы устанавливают в качестве параметров блока.

Допустимый диапазон значений  $T_{вкл}$  и  $T_{откл}$  – от 0,00 с до 24 ч.



**Лист изменений в версиях документа**

Номер версии	Дата выпуска	Содержание основных изменений
01	15.02.2010	Вновь созданное руководство
02	25.02.2010	Дополнено описание интерфейса
03	30.06.2010	Добавлены новые функциональные блоки и функций, описаны появившиеся возможности работать с промежуточными переменными и константами
04	30.08.2010	Внесено описание работы с макросами
05	29.09.2010	Исправлено описание блока SEL
06	07.03.2011	Добавлен раздел «Моделирование работы коммутационной программы» Изменены пределы уставок для ФБ TOF, TON, TP (48 дней) и ФБ BLINK (49 дней)
07	29.05.2011	Внесено описание работы с сетевыми возможностями – работа с сетевыми переменными Изменено описание статус-бара основного окна приложения
08	08.10.2011	Добавлено описание ФБ, работающих с типом «число с плавающей точкой»