



Модуль генерации Web-интерфейсов  
**(SimInTech Generation of Web-GUI)**



## Содержание

СОДЕРЖАНИЕ.....	2
1 НАЗНАЧЕНИЕ МОДУЛЯ И СТРУКТУРНАЯ СХЕМА.....	4
2 СОЗДАНИЕ ПРОЕКТА.....	6
3 СОЗДАНИЕ КОНФИГУРАЦИИ И ГЕНЕРАЦИЯ ВЫХОДНЫХ ФАЙЛОВ.....	8
3.1 Краткое описание процесса создания конфигурации и генерации.....	8
3.2 Работа в окне генератора интерфейса.....	8
3.3 Рекомендуемая структура папок проекта.....	10
4 ОТЛАДКА ИНТЕРФЕЙСОВ.....	12
4.1 Отладка в SimInTech и браузере с помощью блока «Web-сервер».....	12
4.2 Отладка в SimInTech.....	12
5 ЗАПИСЬ ЗНАЧЕНИЙ С WEB-СТРАНИЦЫ.....	14
6 ПОЛЬЗОВАТЕЛЬСКИЕ ОКНА УПРАВЛЕНИЯ (МОДАЛЬНЫЕ ОКНА).....	15
7 ВЕРХНЕЕ И НИЖНЕЕ МЕНЮ.....	17
8 БЛОКИ, ИСПОЛЬЗУЕМЫЕ ДЛЯ ОТОБРАЖЕНИЯ ДАННЫХ НА ИНТЕРФЕЙСАХ	19
8.1 Аналоговый датчик.....	19
8.2 Дискретный.....	23
8.3 Ваг (вертикальный).....	25
8.4 Стрелочный прибор.....	27
8.5 График.....	28
8.6 Кнопка.....	29
8.7 Вывод времени.....	31
8.8 Таблица.....	33
8.9 CheckBox (галочка).....	34
8.10 Переключатель.....	35
8.11 Индикатор с возможностью управления.....	37
8.12 Кнопка управления с автовозвратом.....	38
8.13 Круглая кнопка.....	40
8.14 Ползунок.....	40
8.15 Вращающиеся элементы.....	43
8.16 Воспроизведение звука.....	44
8.17 Блоки теплогидравлики.....	46
9 РАБОТА С НЕСКОЛЬКИМИ КАНАЛАМИ.....	47
9.1 Архитектура системы.....	47
9.2 ПО и конфигурационные файлы, устанавливаемые на контроллерах.....	48
9.3 ПО и конфигурационные файлы, устанавливаемые на компьютере с АРМ.....	49



9.4	Генерация проекта, который будет работать на компьютере с АРМ .....	50
10	СОЗДАНИЕ НОВЫХ БЛОКОВ .....	52
10.1	Основные положения .....	52
10.2	Простой пример блока для вывода значения сигнала .....	55
10.3	Получение свойств блока в скрипте. Изменение фона прямоугольника по условию .....	56
10.4	Работа со статусом сигнала .....	57
10.5	Работа с видимостью графических примитивов.....	58
10.6	Запись значений .....	59
11	ПОЛЕЗНЫЕ НАСТРОЙКИ БРАУЗЕРА .....	61
12	ОРГАНИЗАЦИЯ ПЕРЕХОДА СО СХЕМЫ В WEB-ВЕРСИЮ «TRENDCLIENT WEB» И ОБРАТНО .....	62
12.1	Основные положения .....	62
12.1	Переход в архив по кнопке .....	62
12.1	Переход в архив через блоки «ИНТЕРФЕЙС - Аналоговый датчик» и «ИНТЕРФЕЙС – Дискретный» .....	62
	ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ.....	64



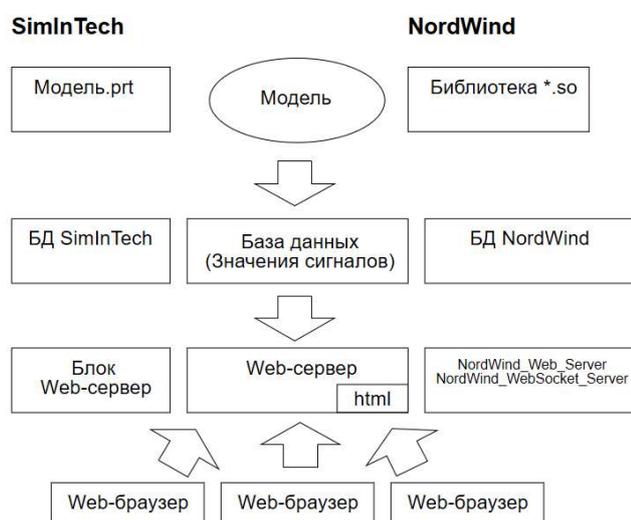
## 1 Назначение модуля и структурная схема

Модуль генерации Web-интерфейсов. Инструмент предназначен для создания анимированных индикаторов и пультов управления на основе Web-сервера. Позволяет создавать Web-GUI для проектов SimInTech и проектов исполняемых в NordWind, обеспечивая подключение к базе данных SimInTech. Web-интерфейсы хранятся в формате HTML файлов и анимируются с помощью JavaScript.

Только проекты, созданные на основе шаблона «Интерфейс» могут быть сгенерированы в html формат. Только специальные блоки из библиотеки «Интерфейс» могут быть использованы для оживления схемы. Любые другие блоки из других библиотек будут неживыми, то есть будут выглядеть как картинка.

В качестве Web-сервера могут выступать следующие программные компоненты:

- Блок «Web-сервер» в SimInTech, который можно установить на схему;
- Web-сервер NordWind\_WebSocket\_Server из набора исполняемой среды NordWind;
- Web-сервер NordWind\_Web\_Server из набора исполняемой среды NordWind;



**Рисунок 1.1 – Структурная схема**

Принцип работы Web-GUI для SimInTech и NordWind одинаковый.

Есть математическая модель (алгоритм). В случае SimInTech она представляет из себя один или несколько проектов prt. В случае NordWind алгоритм кодогенерируется и компилируется в одну или несколько библиотек so, которые подключаются к NordWind.



Модель (алгоритм) работают и результат своей работы в виде значений сигналов помещают в базу данных. В случае SimInTech она представляет из себя стандартную базу данных сигналов. В NordWind есть своя внутренняя базу данных сигналов, через которую различные приложения получают значения сигналов.

Значения сигналов из базы данных забирает Web-сервер. В случае SimInTech это блок «Web-сервер». В NordWind это программы NordWind\_WebSocket\_Server и NordWind\_Web\_Server.

У Web-сервера есть папка html, в которую копируются файлы, созданные генератором.

К Web-серверу подключаются Web-браузеры.

В качестве Web-браузеров необходимо использовать Firefox не ниже версии 70, Google Chrome не ниже версии 74, Microsoft Edge не ниже версии 140, Яндекс Браузер не ниже версии 24.



## 2 Создание проекта

Проект представляет из себя схему, которая будет отображаться в браузере.

Создание проекта происходит через пункт меню «Файл/новый проект/Web Интерфейс».

Новый проект имеет внутри себя прямоугольник. Этот прямоугольник называется «Область видимости». Он отображает пропорции области, которая будет видна в web-браузере. Блоки вне этого прямоугольника не будут видны в браузере. Пропорции можно задать через «параметры расчета» - «Область видимости».

Цвет фона схемы можно менять через меню «Вид – Цвет фона».

После создания проекта поместите на схему блоки для отображения информации с вкладки «Интерфейс». В блоке необходимо задать имя сигнала. Имя сигнала состоит из трех частей: Имя сигнала, символ нижнего подчеркивания, имя хвоста сигнала. Например, есть сигнал с именем КВА00АА3\_xq. В свойстве «Имя сигнала» необходимо задать КВА00АА3, в свойстве «Имя хвоста сигнала» необходимо задать xq.

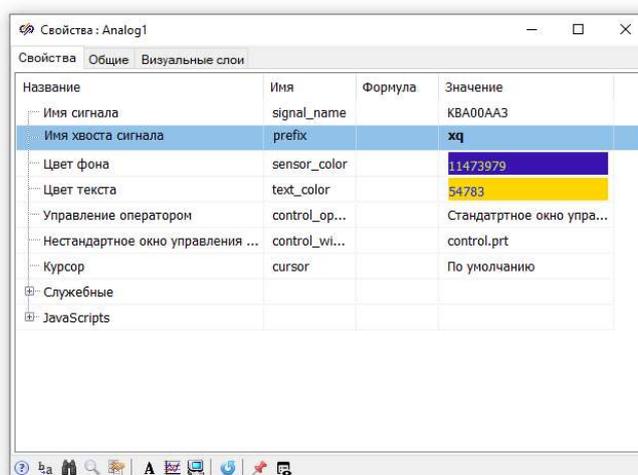


Рисунок 2.1 – Окно задания свойств блока

Для создания кнопки перехода на другую страницу создайте блок «Кнопка». В свойстве «Действие» выберите «Перейти на схему». В свойстве «Схема, на которую перейти» напишите имя проекта, на который будет осуществлен переход при нажатии на кнопку. Например, shema2.prt. В свойстве «Подпись на кнопке» введите текст, который будет отображаться на кнопке.

Можно менять подсказку, появляющуюся при наведении мыши на блок. Для этого измените значение свойства «Подсказка» на вкладке «Общие». По умолчанию в подсказке отображается имя сигнала.



Не используйте для создания графики интерфейсов растровые изображения (png, jpg, bmp, gif и т.д.).

Сохраните проект в папку или в подпапку, где будет сохранена конфигурация. О создании конфигурации в следующих пунктах. В параметрах расчета можно задать свойства проекта.



### 3 Создание конфигурации и генерация выходных файлов

#### 3.1 Краткое описание процесса создания конфигурации и генерации

Процесс создания конфигурации кратко можно описать так:

1. Создать один или несколько проектов схем. Одна схема – один проект. На схемы установить блоки с вкладки «Интерфейс». В свойствах блоков задать имена отображаемых сигналов.
2. Создать конфигурацию. Конфигурация состоит из нескольких схем. В конфигурацию добавить созданные схемы.
3. Настроить выходную директорию на вкладке «Настройки».
4. Запустить генерацию. Сгенерированные файлы из выходной директории скопировать на web-сервер.

#### 3.2 Работа в окне генератора интерфейса

Конфигурация состоит из нескольких проектов.

Работа с конфигурацией происходит в окне «Генератор интерфейса». Для вызова окна в верхней панели SimInTech выберите пункт меню «Инструменты/Генерация интерфейса».

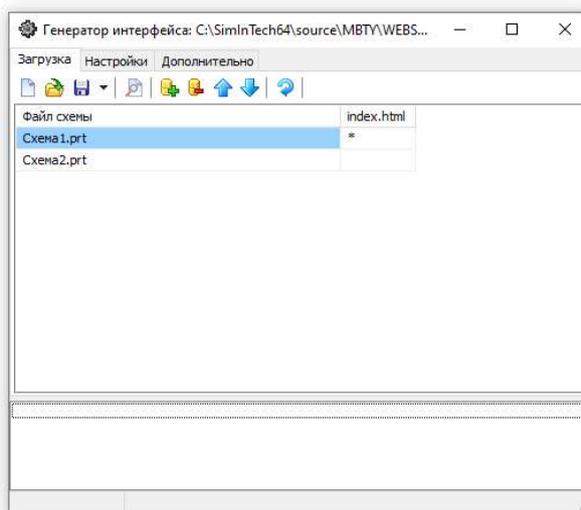


Рисунок 3.2.1 – Окно «Генератор интерфейса»

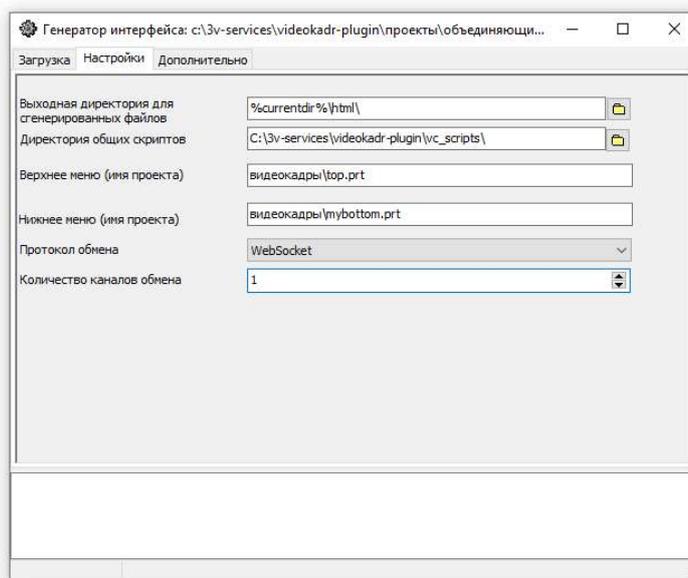
Для создания конфигурации нажмите на кнопку «Новая конфигурация». Сохраните проект нажав на кнопку «Сохранить конфигурацию».



Добавьте один или несколько проектов в конфигурацию. Для добавления проекта нажмите на кнопку «Добавить файлы». Файл добавится в конфигурацию в список.

Для удаления проекта из конфигурации выделите проект и нажмите на кнопку «Удалить файлы».

Перейдите во вкладку «Настройки».



**Рисунок 3.2.2 – Вкладка «Настройки» окна «Генератор интерфейса»**

В поле «Выходная директория для сгенерированных файлов» введите полный путь к папке с выходными файлами. Можно вводить шаблон. Подстрока `%currentdir%` при генерации заменяется на полный путь к файлу конфигурации `vk`. Например, если создать в папке с конфигурацией папку `html`, то в это поле можно написать `%currentdir%\html\`.

В поле «Директория общих скриптов» вводится полный путь к папке `vc_scripts` - `%bin%\vc_scripts\`. Обычно не требует изменений со стороны пользователя, т.к. там по умолчанию уже находится `%bin%\vc_scripts\`. Подстрока `%bin%` при генерации заменяется на папку с программой (допустим, `C:\SimInTech64\bin`). Например, строка `%bin%\vc_scripts\` будет преобразована в `C:\SimInTech64\bin\vc_scripts\`. Папка `vc_scripts` распространяется с `SimInTech`. В ней не нужно ничего менять.

В выпадающем списке «Протокол обмена» выберите способ получения текущих значений сигналов от Web-сервера:

- `http` – периодический запрос значений всех сигналов в Web-сервере по протоколу `http` в формате `JSON`. Интервал обновления данных на Web-странице 1 секунда. Выберите его, если в качестве Web-сервера используется блок Web-сервер в



SimInTech или Web-сервер NordWind\_Web\_Server из набора исполняемой среды NordWind;

- WebSocket – получение значений только тех сигналов, которые есть на Web-странице в рамках одного установленного TCP/IP соединения по протоколу WebSocket. Данные передаются по изменению сигнала (спорадически). Этим достигается более быстрое обновление данных на Web-странице, чем http. Выберите его, если в качестве Web-сервера используется Web-сервер NordWind\_WebSocket\_Server из набора исполняемой среды NordWind.

Для генерации выходных файлов перейдите во вкладку «Загрузка» и нажмите на кнопку «Сгенерировать файлы» . Файлы окажутся в папке, которую Вы указали в поле «Выходная директория для сгенерированных файлов». Загрузите эти файлы в папку html Web-сервера. Если используется блок Web Server в SimInTech, то загрузите эти файлы в папку, указанную в свойстве «Имя папки с файлами (folder)» блока. Файлы следует загружать с заменой существующих файлов. Если в папке уже есть файлы от предыдущего проекта, то сначала удалите старые файлы, а потом загрузите новые.

В адресной строке браузера наберите адрес и порт Web-сервера. Например, <http://192.168.56.112:8088>. Отобразится загруженная схема.

Изменения, которые Вы вносите в проект, Вы можете увидеть в браузере после загрузки файлов на Web-сервер и обновления страницы.

Если Вы внесли незначительные изменения в один проект, которые никак не могут затронуть другие проекты. Например, поменяли цвет у блока. Вы можете выполнить генерацию файлов только этого проекта. Для этого нажмите на кнопку «Сгенерировать файлы только выделенного проекта» .

### 3.3 Рекомендуемая структура папок проекта

Имя	Дата изменения	Тип	Размер
html	04.10.2025 11:50	Папка с файлами	
Верхнее меню	04.10.2025 11:50	Папка с файлами	
Окна управления	04.10.2025 11:50	Папка с файлами	
Схемы	04.10.2025 11:50	Папка с файлами	
db.db	10.09.2025 9:06	Data Base File	1 КБ
db.dbconf	10.09.2025 9:08	Файл "DBCONF"	3 КБ
model.mgr	10.09.2025 9:08	Файл "MGR"	1 КБ
model.prt	10.09.2025 9:08	Файл "PRT"	10 КБ
тест.vk	23.08.2025 12:51	Файл "VK"	1 КБ

Рисунок 3.3.1 – Папка с проектом



Конфигурация может содержать много файлов. Чтобы эти файлы не были свалены в одной папке длинным списком, рекомендуется раскладывать их по разным папкам. Например, на рисунке 3.3.1 схемы находятся в папке «Схемы», Окна управления находятся в папке «Окна управления». Пути к файлам в настройках задаются относительно файла конфигурации (файл с расширением vk). Папка, где находится файл конфигурации считается корневой. Например, для задания файла окна управления в настройках следует писать «Окна управления/окно управления.prt». Т.к. файл «окно управления.prt» находится в папке «Окна управления». А папка «Окна управления» уже находится в корневой папке. Т.е. той, где находится файл конфигурации с расширением vk.



## 4 Отладка интерфейсов

### 4.1 Отладка в SimInTech и браузере с помощью блока «Web-сервер»

Отладка в SimInTech позволяет посмотреть созданные схемы со значениями сигналов в браузере без необходимости загрузки на контроллер. В качестве Web-сервера будет выступать блок «Web-сервер».

Создайте пакет, если его нет, и добавьте в него проекты с алгоритмами, которые будут загружены на контроллер. Значения сигналов от проектов должны передаваться в базу данных. Создайте проект, включите его в пакет, настройте базу данных такую же как и в проекте с алгоритмами. Добавьте в проект блок «Web-сервер» с вкладки «Интерфейс». Чтобы блок «Web-сервер» брал все значения из базы данных установите свойство блока «Передавать все сигналы из базы данных» в «Да». В свойстве «Имя папки с файлами» блока укажите папку с выходными файлами. Папка, указанная в свойстве «Имя папки с файлами» должна быть в одной папке с проектом с блоком. В свойстве «Порт» укажите порт Web-сервера, например 8088. В адресной строке браузера наберите адрес и порт Web-сервера. Например, <http://127.0.0.1:8088>, если браузер запущен на этом же компьютере. Отобразится схема.

Изменения, которые Вы вносите в проект, Вы можете увидеть в браузере после генерации и копирования файлов в папку Web-сервера и обновления страницы.

Вы можете настроить выходную директорию для сгенерированных файлов такую же как у блока «Web-сервер». Тогда не нужно будет копировать файлы.

### 4.2 Отладка в SimInTech

Режимом отладки является режим, когда в проекте «Интерфейс» можно просматривать значения сигналов непосредственно в SimInTech на этапе разработки.

«Интерфейс» в таком режиме берет значения сигналов из базы данных.

Добавьте проект «Интерфейс» в пакет с математической моделью. В настройках проекта укажите базу данных, с которой работает пакет. Запустите пакет на расчет. «Интерфейс» будет отображать значения сигналов из базы данных.

В режиме отладки работают только блоки «Аналоговый датчик», «Дискретный», «Ваг», «Стрелочный прибор». График и таблица выглядит в виде прямоугольника.



В режиме отладки пользовательские модальные окна не отображают значения из базы данных.

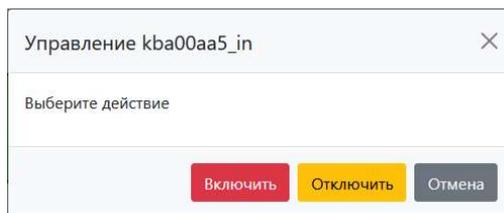
В режиме отладки не работает диалог выбора нескольких нестандартных окон управления. В режиме отладки не отображаются статусы и метки времени сигналов.



## 5 Запись значений с Web-страницы

С помощью блоков «Дискретный» и «Аналоговый датчик» можно записывать значения в сигналы (выдавать команды управления). Для этого в свойстве блока «Управление оператором» выберите пункт «Стандартное окно управления». В браузере при клике мышкой на блоке будет появляться окно управления.

Для блока «Дискретный» окно имеет следующий вид:

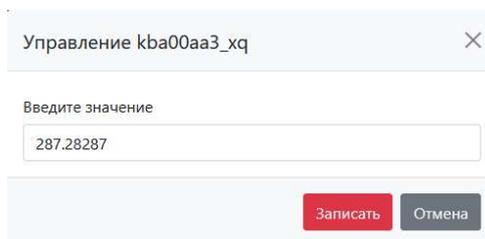


**Рисунок 5.1 – Окно управления блока «Дискретный»**

При нажатии на кнопку «Включить» в сигнал записывается значение 1. При нажатии на кнопку «Отключить» в сигнал записывается значение 0. Для закрытия окна нажмите на кнопку «Отмена» или крестик в верхнем правом углу появившегося окна.

Аналог этого окна можно создать самостоятельно с помощью пользовательского окна управления и кнопок.

Для блока «Аналоговый датчик» окно имеет следующий вид:



**Рисунок 5.2 – Окно управления блока «Аналоговый датчик»**

В поле «Введите значение» напишите новое значение сигнала. При нажатии на кнопку «Записать» или клавишу Enter в сигнал записывается введенное значение. Для закрытия окна нажмите на кнопку «Отмена» или крестик в верхнем правом углу появившегося окна или клавишу escape.



## 6 Пользовательские окна управления (модальные окна)

Модальное окно – это окно, которое появляется при клике мышкой по блоку. Модальное окно служит для отображения значений сигналов, у которых в названии начало имени сигнала (имя до нижнего подчеркивания) одинаковое. Модальное окно позволяет отображать однотипную информацию. Например, на схеме может быть несколько однотипных задвижек, а окно для отображения информации с задвижки можно создать одно.

Для создания модального окна создайте новую схему Web-Интерфейс. Эта схема будет модальным окном. Расположите на схеме блоки. В свойствах блоков заполните «Имя хвоста сигнала». Значение свойства «Имя сигнала» может быть любым. Сохраните схему. Например, с именем control.prt.

Откройте схему, из которой будет вызываться модальное окно. Выделите блок, по клику на который, должно появляться модальное окно. В свойстве блока «Управление оператором» выберите «Нестандартное окно управления», в свойстве «Нестандартное окно управления (имя проекта)» введите имя проекта, который будет модальным окном (control.prt). Если файл control.prt лежит в папке относительно файла конфигурации с расширением vk, то вводить следует имя\_папки/control.prt.

Итоговое имя сигнала блока, размещенного в модальном окне, вычисляется следующим образом: имя сигнала до нижнего подчеркивания будет браться из значения свойства «Имя сигнала» блока, из которой вызывается модальное окно; имя сигнала после нижнего подчеркивания будет браться из значения свойства «Имя хвоста сигнала» блока, размещенного в модальном окне.

Для задания пропорций окна управления так же предназначена «область видимости».

Ширина окна управления задаётся в свойстве «Ширина окна управления» секции «Окно управления». Если выбрано значение свойства «Берется из области видимости (в пикселях)», то размеры окна управления в пикселях берутся из значений свойств «Область видимости (ширина и высота)». Например, если в области видимости задана ширина 300 и высота 200, то ширина окна управления будет 300 пикселей, а высота 200 пикселей на любом разрешении монитора. Если размер окна управления должен быть пропорционален экрану браузера, то следует выбирать следующие значения свойств. Например, «1/2 ширины окна браузера». В этом случае окно управления будет занимать половину экрана по ширине на любом разрешении монитора.



Если выбрано значение свойства «Задать вручную», то ширина окна управления задаётся в пикселях. Она берётся из параметров расчета свойства «Ширина окна управления в пикселях», если там задан не ноль. Если там задан ноль, то ширина берётся из фактической ширины окна проекта.

Окна управления появляются поверх блока, по которому кликнул пользователь, если выбрано значение «Поверх блока, на котором кликнули» в свойстве «Где показывать окна управления?». Если выбрано значение «Посередине вверху», то окна управления появляются посередине вверху схемы. Свойство нужно устанавливать в проекте схемы, а не в проекте окна управления. Свойство применяется для всех окон управления, появляющихся на этой схеме.

Дополнительные свойства окна управления задаются параметрах расчета, в секции «Окно управления».



## 7 Верхнее и нижнее меню

Верхнее меню служит для размещения на нем кнопок перехода на схемы и появляется на всех сгенерированных web-страницах вверху.

Нижнее меню служит для размещения на нем кнопок перехода на схемы и появляется на всех сгенерированных web-страницах внизу.

Для создания верхнего или нижнего меню создайте новую схему Интерфейс. Измените пропорции «Области видимости», чтобы ширина была больше высоты. Разместите на схеме кнопки перехода (Блоки «Кнопка»). В свойстве блока «Схема, на которую перейти» напишите имя проекта, на который будет осуществлен переход при нажатии на кнопку. В свойстве «Подпись на кнопке» введите текст, который будет отображаться на кнопке. Сохраните проект, например с именем top.prt.

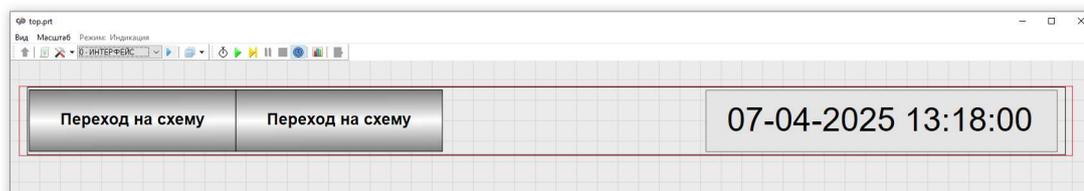


Рисунок 7.1 – Окно проекта верхнего меню

В окне конфигурации перейдите во вкладку «Настройки». В поле «Верхнее меню (имя проекта)» введите имя проекта меню. В нашем случае top.prt. Если файл top.prt лежит в папке относительно файла конфигурации с расширением vk, то вводить следует имя\_папки/top.prt.

Сгенерируйте выходные файлы и загрузите их на сервер. После обновления страницы должно появиться верхнее меню.



Рисунок 7.2 – Окно браузера с верхним меню



Для настройки нижнего меню в поле «Нижнее меню (имя проекта)» введите имя проекта меню.



## 8 Блоки, используемые для отображения данных на Интерфейсах

### 8.1 Аналоговый датчик

**213.2**

- отображение блока «Аналоговый датчик» на схеме.

Блок «Аналоговый датчик» предназначен для вывода значения аналогового сигнала на схему. По двойному щелчку левой кнопкой мыши на блоке осуществляется отображение диалогового окна. На рисунке 3.1.1 представлен диалог для задания свойств блока.

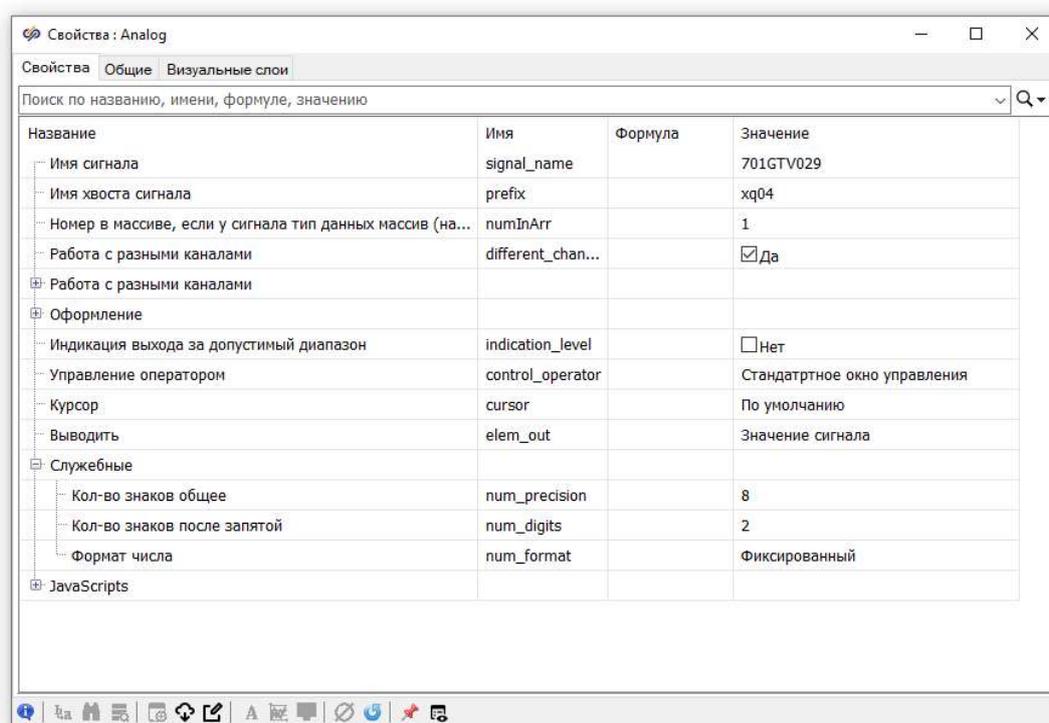


Рисунок 8.1.1 – Диалоговое окно блока «Аналоговый датчик»

Перечень свойств блока приведен в таблице 8.1.1:

№ п/п	Название свойства	Описание свойства
1.	Имя сигнала	Имя сигнала до нижнего подчеркивания.
2.	Имя хвоста сигнала	Имя сигнала после нижнего подчеркивания.
3.	Номер в массиве, если у сигнала тип данных массив	Значение одного сигнала может быть представлено в виде массива. Можно вывести один элемент из массива. Нумерация начинается с 1.
4.	Работа с разными каналами	Если «Да», то позволяет выводить информацию о различиях значений и статусов по разным каналам.
5.	Номер канала, из которого отображать данные	0 – объединять значения и статусы из разных каналов. Если задать номер канала, то будут отображаться данные только из этого канала.



6.	Допустимая величина расхождения значений каналов	Если значения по каналам отличаются более, чем на эту величину, блок окрашивается в другой цвет.
7.	Цвет фона при расхождении значений каналов	Если значения по каналам отличаются более, чем на величину, заданной в свойстве «Допустимая величина расхождения значений каналов», фон окрашивается в этот цвет.
8.	Цвет текста при расхождении значений каналов	Если значения по каналам отличаются более, чем на величину, заданной в свойстве «Допустимая величина расхождения значений каналов», текст окрашивается в этот цвет.
9.	Цвет рамки при расхождении статусов каналов	Если значения по каналам разные, рамка окрашивается в этот цвет.
10.	Стиль отображения	Варианты: - Одна строка - Две строки. Во вторую строку выводится единица измерения.
11.	Текст подписи	Подпись, которая выводится перед числовым значением. Например, P=.
12.	Цвет фона в норме	Цвет фона блока.
13.	Цвет текста в норме	Цвет текста блока.
14.	Менять цвет фона при плохом статусе	Если выбрано «Да», при плохом статусе цвет фона окрашивается в цвет, заданный в свойстве «Цвет фона при плохом статусе».
15.	Цвет рамки плохого статуса	Если плохой статус, вокруг блока появляется рамка этого цвета.
16.	Цвет линии рамки	Цвет линии прямоугольника вокруг блока.
17.	Наличие фона	Если значение «Да», то значение отображается в залитом фоновом прямоугольнике.
18.	Мигание фона при неактивности канала связи	Если выбрано «Да», будет мигать фон при неактивности канала связи.
19.	Индикация выхода за допустимый диапазон	Если поставить эту галочку, то можно задать диапазоны значений, при выходе за которые будет меняться цвет фона и цвет текста
20.	Источник информации о выходе за уставку	Варианты: - Свойства. Для индикации значения аварийного и предупредительного берутся из свойств блока; - Другой сигнал. Для индикации значения информация о выходе за уставку берется из сигнала, заданном в свойстве «Имя хвоста сигнала выхода за уставку». Значения этого сигнала: 0 – норма, 1 – предупредительное, 2 – аварийное.



21.	Верхнее аварийное значение, Верхнее предупредительное значение, Нижнее предупредительное значение, Нижнее аварийное значение	Значения, при выходе за которые будет меняться цвет фона и цвет текста
22.	Цвет фона аварийного значения	Цвет фона, который будет, если значение меньше нижнего аварийного значения или значение больше верхнего аварийного значения
23.	Цвет текста аварийного значения	Цвет текста, который будет, если значение меньше нижнего аварийного значения или значение больше верхнего аварийного значения
24.	Цвет фона предупредительного значения	Цвет фона, который будет, если значение меньше нижнего предупредительного значения или значение больше верхнего предупредительного значения
25.	Цвет текста предупредительного значения	Цвет текста, который будет, если значение меньше нижнего предупредительного значения или значение больше верхнего предупредительного значения
26.	Управление оператором	<p>Варианты выбора:</p> <ul style="list-style-type: none"> <li>- Нет – запись нового значения сигнала невозможна;</li> <li>- Стандартное окно управления – при нажатии на блок появляется окно, в котором можно ввести новое числовое значение и записать его в NordWind.</li> <li>- Нестандартное окно управления - при нажатии на блок появляется окно, созданное пользователем. Имя проекта окна задаётся в свойстве «Нестандартное окно управления».</li> <li>- Несколько нестандартных окон управления. Появляется окно с выбором из нескольких вариантов. Варианты нужно задать в свойстве «Нестандартные окна управления» в следующем формате:</li> </ul> <pre>sensor h lim.prt;Ввод уставки H LIM для датчика \$self.signal_name\$ sensor hh lim.prt;Ввод уставки HH LIM для датчика \$self.signal_name\$</pre> <p>Сначала идет имя проекта окна управления. Затем идет строка, которую видит пользователь. Строка \$self.signal_name\$ заменяется на имя сигнала (до нижнего подчеркивания).</p>
27.	Нестандартное окно управления (имя проекта)	Имя проекта пользовательского окна управления. Это окно появляется при клике мышкой на блоке и значении свойства «Управление оператором» - «Нестандартное окно управления».
28.	Курсор	<p>Вид курсора мыши при наведении на блок в web-браузере.</p> <p>Варианты выбора:</p> <ul style="list-style-type: none"> <li>- По умолчанию – если есть возможность записи нового значения по клику на блоке, то курсор будет в виде руки. Если возможность записи нет, то курсор будет таким, каким его установит браузер;</li> </ul>



		- Всегда стрелка - курсор будет таким, каким его установит браузер; - Всегда рука - курсор будет в виде руки.
29.	Выводить	Варианты выбора: - Значение сигнала; - Статус сигнала; - Время сигнала; - Имя сигнала. Выводится, то, что задано в свойстве «Имя сигнала»; - Строка в зависимости от значения. Выводится текст, который меняется в зависимости от значения. Это задаётся в свойствах «Список строк в зависимости от значения» и «Строка по умолчанию»
30.	Список строк в зависимости от значения	Выводится текст, который меняется в зависимости от значения сигнала. Формат: 1;вкл;#7d8471 2;откл;#ab1d70 4;авария Сначала идет значение, затем текст, затем цвет при необходимости.
31.	Строка по умолчанию	Выводится текст, при значении сигнала, которое не попадает в варианты, описанные в предыдущем свойстве.
32.	Кол-во знаков общее	Общее количество знаков при отображении числового значения.
33.	Кол-во знаков после запятой	Количество знаков после запятой.
34.	Формат числа	Выбор формата отображения динамического числового значения. Варианты выбора: - Основной; - Экспоненциальный; - Фиксированный.

Можно менять подсказку, появляющуюся при наведении мыши на блок. Для этого измените значение свойства «Подсказка» на вкладке «Общие». По умолчанию в подсказке отображается имя сигнала. В режиме пользовательских модальных окон в подсказке отображается имя сигнала будет неверным, т.к. оно меняется динамически. Поэтому, либо значение свойства «Подсказка» оставляйте пустое, либо напишите значение свойства, не зависящее от имени сигнала.



## 8.2 Дискретный



- отображение блока «Дискретный» на схеме.

Блок «Дискретный» предназначен для вывода значения дискретного сигнала на схему. Блок может отображать только 2 значения: 1 или 0 (включен или отключен). По двойному щелчку левой кнопкой мыши на блоке осуществляется отображение диалогового окна. На рисунке 8.2.1 представлен диалог для задания свойств блока.

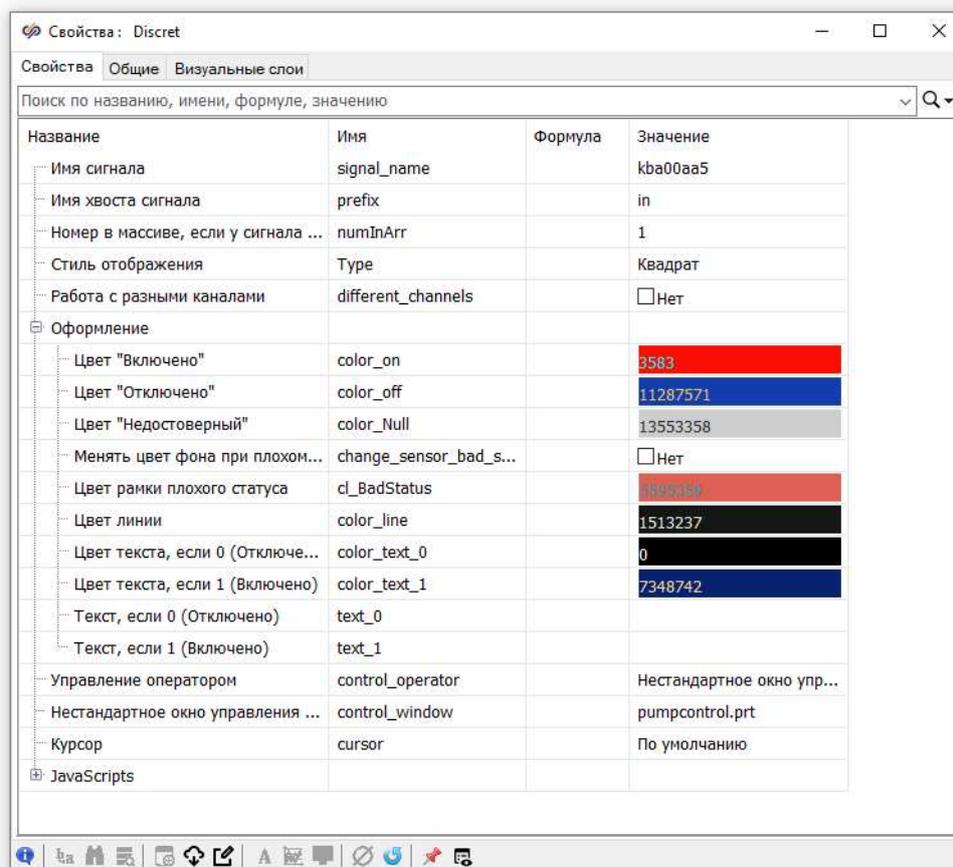


Рисунок 8.2.1 – Диалоговое окно блока «Дискретный»

Перечень свойств блока приведен в таблице 8.2.1:

№ п/п	Название свойства	Описание свойства
1.	Имя сигнала	Имя сигнала до нижнего подчеркивания.
2.	Имя хвоста сигнала	Имя сигнала после нижнего подчеркивания.
3.	Номер в массиве, если у сигнала тип данных массив	Значение одного сигнала может быть представлено в виде массива. Можно вывести один элемент из массива. Нумерация начинается с 1.
4.	Стиль отображения	Варианты:



		- Квадрат - Круг - Круг с градиентом
5.	Работа с разными каналами	Если «Да», то позволяет выводить информацию о различиях значений и статусов по разным каналам.
6.	Номер канала, из которого отображать данные	0 – объединять значения и статусы из разных каналов. Если задать номер канала, то будут отображаться данные только из этого канала.
7.	Цвет фона при расхождении значений каналов	Если значения по каналам отличаются, фон окрашивается в этот цвет.
8.	Цвет рамки при расхождении статусов каналов	Если значения по каналам разные, рамка окрашивается в этот цвет.
9.	Цвет "Включено"	Цвет, в который окрашивается фон блока, когда значение сигнала равно 1.
10.	Цвет "Отключено"	Цвет, в который окрашивается фон блока, когда значение сигнала равно 0.
11.	Цвет "Недостовверный"	Цвет, в который окрашивается фон блока, когда значение сигнала недостоверно. Например, когда нет связи между web-браузером и сервером.
12.	Менять цвет фона при плохом статусе	Если «Да», то при плохом статусе цвет фона окрашивается в цвет, заданный в свойстве «Цвет фона при плохом статусе».
13.	Цвет рамки плохого статуса	Если плохой статус, вокруг блока появляется рамка этого цвета.
14.	Цвет линии	Цвет линии
15.	Цвет текста, если 0 (Отключено)	Цвет, в который окрашивается текст, если значение сигнала равно 0
16.	Цвет текста, если 1 (Включено)	Цвет, в который окрашивается текст, если значение сигнала равно 1
17.	Текст, если 0 (Отключено)	Текст, который выводится на блоке, если значение сигнала равно 0
18.	Текст, если 1 (Включено)	Текст, который выводится на блоке, если значение сигнала равно 1
19.	Управление оператором	Варианты выбора: - Нет – запись нового значения сигнала невозможна; - Стандартное окно управления – при нажатии на блок появляется окно с кнопками «Включить» и «Отключить». При нажатии на кнопку «Включить» в NordWind записывается значение 1. При нажатии на кнопку «Отключить» в NordWind записывается значение 0. - Нестандартное окно управления - при нажатии на блок появляется окно, созданное пользователем.



		<p>Имя проекта окна задаётся в свойстве «Нестандартное окно управления».</p> <p>- Несколько нестандартных окон управления. Появляется окно с выбором из нескольких вариантов. Варианты нужно задать в свойстве «Нестандартные окна управления» в следующем формате:</p> <pre>sensor_h_lim.prt;Ввод уставки Н LIM для датчика \$self.signal_name\$ sensor_hh_lim.prt;Ввод уставки НН LIM для датчика \$self.signal_name\$</pre> <p>Сначала идет имя проекта окна управления. Затем идет строка, которую видит пользователь. Строка \$self.signal_name\$ заменяется на имя сигнала (до нижнего подчеркивания).</p>
20.	Нестандартное окно управления (имя проекта)	<p>Имя проекта пользовательского окна управления. Это окно появляется при клике мышкой на блоке и значении свойства «Управление оператором» - «Нестандартное окно управления».</p>
21.	Курсор	<p>Вид курсора мыши при наведении на блок в web-браузере.</p> <p>Варианты выбора:</p> <ul style="list-style-type: none"> <li>- По умолчанию – если есть возможность записи нового значения по клику на блоке, то курсор будет в виде руки. Если возможность записи нет, то курсор будет таким, каким его установит браузер;</li> <li>- Всегда стрелка - курсор будет таким, каким его установит браузер;</li> <li>- Всегда рука - курсор будет в виде руки.</li> </ul>

Можно менять подсказку, появляющуюся при наведении мыши на блок. Для этого измените значение свойства «Подсказка» на вкладке «Общие». По умолчанию в подсказке отображается имя сигнала. В режиме пользовательских модальных окон в подсказке отображается имя сигнала будет неверным, т.к. оно меняется динамически. Поэтому, либо значение свойства «Подсказка» оставляйте пустое, либо напишите значение свойства, не зависящее от имени сигнала.

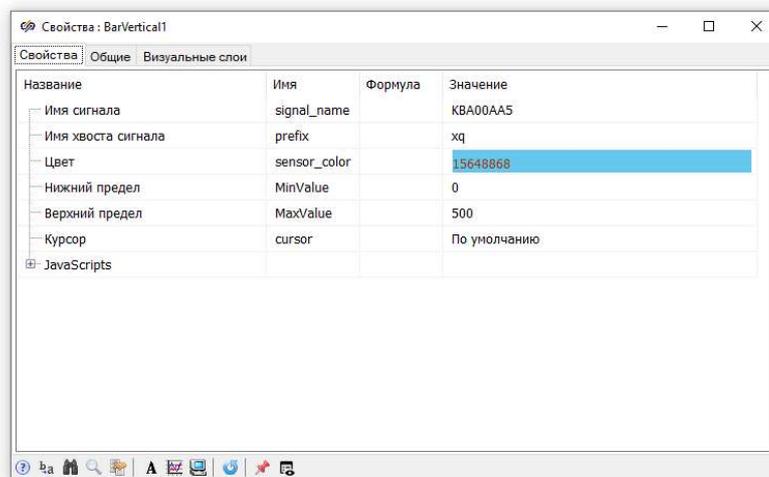
### 8.3 Вар (вертикальный)

 - отображение блока «Вар (вертикальный)» на схеме.

Блок «Вар (вертикальный)» предназначен для вывода значения аналогового сигнала на схему. Блок отображает значение в виде уровня. По двойному щелчку левой кнопкой



мышью на блоке осуществляется отображение диалогового окна. На рисунке 8.3.1 представлен диалог для задания свойств блока.



**Рисунок 8.3.1 – Диалоговое окно блока «Bar (вертикальный)»**

Перечень свойств блока приведен в таблице 8.3.1:

№ п/п	Название свойства	Описание свойства
1.	Имя сигнала	Имя сигнала до нижнего подчеркивания.
2.	Имя хвоста сигнала	Имя сигнала после нижнего подчеркивания.
3.	Цвет	Цвет области, которая визуализирует значение в виде уровня.
4.	Нижний предел	Минимальное значение сигнала, при котором уровень будет у дна.
5.	Верхний предел	Максимальное значение сигнала, при котором уровень будет максимален. Прямоугольник будет полностью заполнен.
6.	Курсор	Вид курсора мыши при наведении на блок в web-браузере. Варианты выбора: - По умолчанию – курсор будет таким, каким его установит браузер; - Всегда стрелка - курсор будет таким, каким его установит браузер; - Всегда рука - курсор будет в виде руки.



## 8.4 Стрелочный прибор



- отображение блока «Стрелочный прибор» на схеме.

Блок «Стрелочный прибор» предназначен для вывода значения аналогового сигнала на схему. Блок отображает значение выводится посредством стрелки-определителя и градуированной шкалы. По двойному щелчку левой кнопкой мыши на блоке осуществляется отображение диалогового окна. На рисунке 8.4.1 представлен диалог для задания свойств блока.

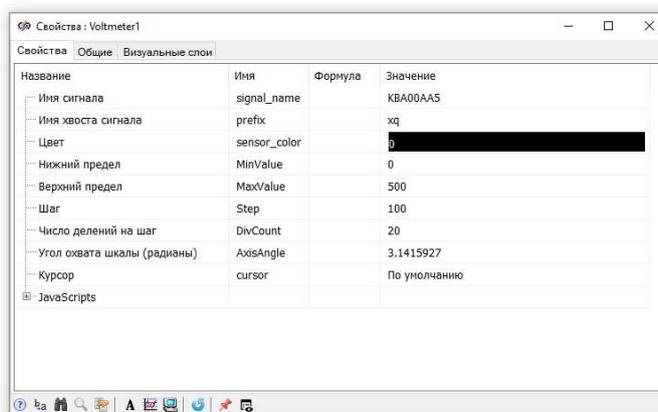


Рисунок 8.4.1 – Диалоговое окно блока «Стрелочный прибор»

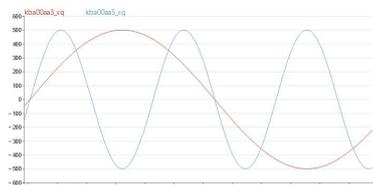
Перечень свойств блока приведен в таблице 8.4.1:

№ п/п	Название свойства	Описание свойства
1.	Имя сигнала	Имя сигнала до нижнего подчеркивания.
2.	Имя хвоста сигнала	Имя сигнала после нижнего подчеркивания.
3.	Цвет	Цвет шкалы.
4.	Нижний предел	Начальное значение шкалы.
5.	Верхний предел	Максимальное значение шкалы.
6.	Шаг	Размер промежутка в единицах шкалы для отображения цифровых подписей над рисками шкалы.
7.	Число делений на шаг	Число отображаемых рисок шкалы между соседними цифровыми подписями, включая риски с подписями.
8.	Угол охвата шкалы (радианы)	Угол сектора в радианах, в который будет вписана шкала. Изменяется симметрично относительно вертикальной оси.



9.	Курсор	<p>Вид курсора мыши при наведении на блок в web-браузере.</p> <p>Варианты выбора:</p> <ul style="list-style-type: none"> <li>- По умолчанию – курсор будет таким, каким его установит браузер;</li> <li>- Всегда стрелка - курсор будет таким, каким его установит браузер;</li> <li>- Всегда рука - курсор будет в виде руки.</li> </ul>
----	--------	---

## 8.5 График



- отображение блока «График» на схеме в web-браузере.

Блок «График» предназначен для построения движущегося графика в web-браузере. Блок строит график в темпе процесса и не берет архивных данных. Построение графика начинается с момента открытия схемы в web-браузере. В режиме отладки не работает. При рисовании на схеме выглядит в виде прямоугольника. По двойному щелчку левой кнопкой мыши на блоке осуществляется отображение диалогового окна. На рисунке 8.5.1 представлен диалог для задания свойств блока.

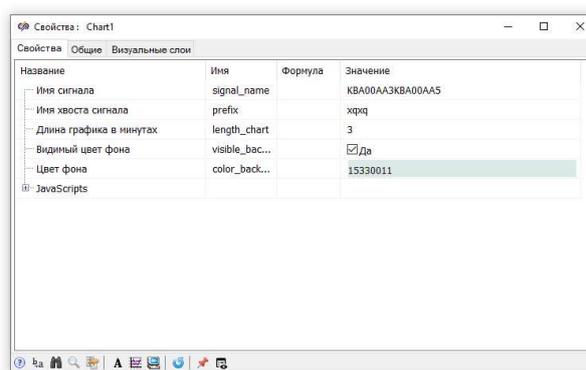


Рисунок 8.5.1 – Диалоговое окно блока «График»

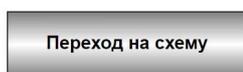
Перечень свойств блока приведен в таблице 8.5.1:

№ п/п	Название свойства	Описание свойства
1.	Имя сигнала	Имена сигналов до нижнего подчеркивания. В окне редактирования свойства вводятся столбиком.



2.	Имя хвоста сигнала	Имена сигналов после нижнего подчеркивания. В окне редактирования свойства вводятся столбиком.
3.	Длина графика в минутах	Интервал времени, отображаемый на графике. Более поздние данные уходят влево и там уничтожаются.
4.	Видимый цвет фона	Если поставлена галочка, то фон графика виден. Если галочка снята, то видны только линии.
5.	Цвет фона	Цвет фона графика.

## 8.6 Кнопка



- отображение блока «Кнопка» на схеме в web-браузере.

Блок «Кнопка» предназначен для организации перехода на другую схему, записи значений и прочих действий в web-браузере и выглядит как кнопка. В режиме отладки не работает. По двойному щелчку левой кнопкой мыши на блоке осуществляется отображение диалогового окна. На рисунке 8.6.1 представлен диалог для задания свойств блока.

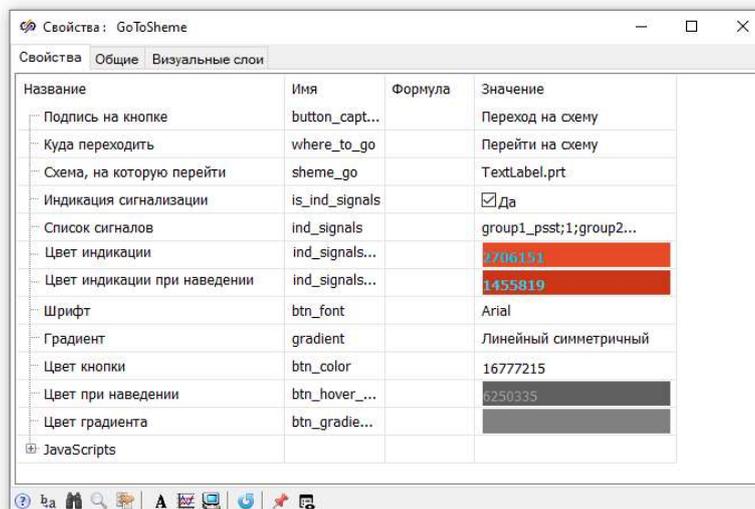


Рисунок 8.6.1 – Диалоговое окно блока «Переход на схему»

Перечень свойств блока приведен в таблице 8.6.1:

№ п/п	Название свойства	Описание свойства
1.	Подпись на кнопке	Текст, который будет отображаться на кнопке.



		<p>Варианты выбора:</p> <ol style="list-style-type: none"> <li>1. Перейти на схему – при нажатии на кнопку будет осуществлен переход на проект, который задан в свойстве «Схема, на которую перейти»;</li> <li>2. На предыдущую страницу (назад) – переход на предыдущий проект;</li> <li>3. На следующую страницу (вперед) – переход на следующий проект;</li> <li>4. Перейти по ссылке – переход по ссылке, заданной в свойстве «Ссылка, на которую перейти»;</li> <li>5. Всплывающее меню – при нажатии на кнопку появляется всплывающее меню;</li> <li>6. Записать значение и/или статус - при нажатии на кнопку происходит запись значения в сигнал. Дополнительные настройки задаются в свойстве «Параметр для записи». Значение для записи задаётся в свойстве «Значение для записи». Статус для записи задаётся в свойстве «Статус для записи». Если после записи нужно, чтобы окно управления с кнопкой закрылось, введите «Да» в свойстве «Закрыть окно управления после записи значения/статуса»;</li> <li>7. Закрыть окно управления - при нажатии на кнопку происходит закрытие окна управления. Имеет смысл только, если кнопка установлена на окно управления.</li> <li>8. Закрыть браузер - при нажатии на кнопку происходит закрытие вкладки браузера. Работает только в Firefox. Чтобы это работало, необходимо настроить через about:config. Свойство dom.allow_scripts_to_close_windows установить в true;</li> <li>9. Закрыть окно управления с контролем разности значений - при нажатии на кнопку происходит закрытие окна управления. Если значения сигналов по каналам, хвосты которых заданы в свойстве «Сигналы, различные значения которых нужно контролировать» различаются на величину большую, чем задано в свойстве «Допустимая разница в значениях», то выдается сообщение.</li> </ol>
2.	Действие	
3.	Схема, на которую перейти	Имя проекта, на который будет осуществлен переход при нажатии на кнопку. Нужно писать название проекта с расширением. Например, Label.prt.
4.	Шрифт	Шрифт текста на кнопке.
5.	Ссылка, на которую перейти	Можно задать произвольную ссылку, на которую будет осуществлен переход при нажатии на кнопку. Свойство появляется, если в свойстве «Куда переходить» выбран вариант «Перейти по ссылке».
6.	Всплывающее меню	Конфигурация меню задаётся в виде текста, разделенного точкой с запятой. Формат: Подпись; проект, на который нужно перейти Пример: Главная страница; Главная.prt Кнопка; Кнопка2.prt



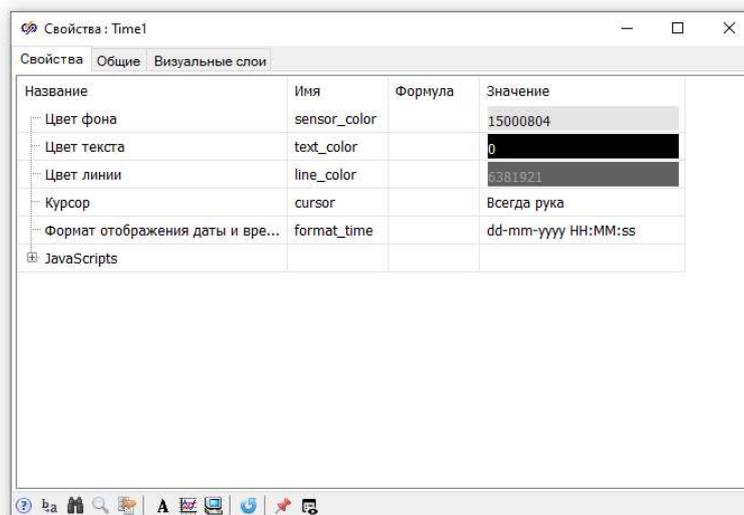
7.	Стиль меню	Варианты: - Светлый; - Темный.
8.	Индикация сигнализации	Кнопка окрашивается в выбранный цвет, если значение хотя бы одного из сигналов равно заданному.
9.	Список сигналов	Список сигналов, при значении хотя бы одного из них заданному, кнопка окрашивается в выбранный цвет. Задаётся в виде текста столбиком следующим способом: Имя_сигнала;значение_сигнала_при_котором_кнопка_окрашивается Пример: group2_psst;1; group3_psst;1; group4_psst;1;
10.	Цвет индикации	Цвет, в который окрашивается кнопка, если значение хотя бы одного из сигналов равно заданному
11.	Цвет индикации при наведении	Цвет, в который окрашивается кнопка при наведении на неё мыши, если значение хотя бы одного из сигналов равно заданному
12.	Градиент	Вид градиента отображения кнопки
13.	Цвет кнопки	Цвет кнопки
14.	Цвет при наведении	Цвет, в который окрашивается кнопка при наведении на неё мыши
15.	Цвет градиента	Цвет градиента

## 8.7 Вывод времени

26-05-2024 20:15:42

- отображение блока «Вывод времени» на схеме в web-браузере.

Блок «Вывод времени» предназначен для вывода текущего времени и даты компьютера, на котором запущен web-браузер. По двойному щелчку левой кнопкой мыши на блоке осуществляется отображение диалогового окна. На рисунке 8.7.1 представлен диалог для задания свойств блока.



**Рисунок 8.7.1 – Диалоговое окно блока «Вывод времени»**

Перечень свойств блока приведен в таблице 8.7.1:

№ п/п	Название свойства	Описание свойства
1.	Цвет фона	Цвет фона.
2.	Цвет текста	Цвет текста.
3.	Цвет линии	Цвет линии.
4.	Курсор	Вид курсора мыши при наведении на блок в web-браузере. Варианты выбора: - По умолчанию – курсор будет таким, каким его установит браузер; - Всегда стрелка - курсор будет таким, каким его установит браузер; - Всегда рука - курсор будет в виде руки.
5.	Формат отображения даты и времени	Строка форматирования даты и времени. Следующие строки символа форматирования могут использоваться в формируемой строке: uuuu = Год из 4-х цифр mm = Номер месяца как 2 цифры dd = Число как 2 цифры HH = Час как 2 цифры MM = Минуты как 2 цифры ss = Секунды как 2 цифры



## 8.8 Таблица

Время	Описание
21-03-25 13:30:12.080	Ошибка связи с контроллером CAN 1 КПИ-Ф канал А
21-03-25 13:30:12.080	Ошибка связи с контроллером CAN 1 КПИ-Ф канал В
21-03-25 13:30:05.565	Ошибка связи с Флеш накопителем 4 КПИ-Ф канал А
21-03-25 13:30:05.565	Ошибка связи с Флеш накопителем 4 КПИ-Ф канал В
21-03-25 13:30:29.709	Ошибка связи с абонентами RS485 канала 1 КПИ-Ф канал А
21-03-25 13:30:29.709	Ошибка связи с абонентами RS485 канала 1 КПИ-Ф канал В
21-03-25 13:30:29.709	Ошибка выдачи реленой команды КПИ-Ф канал А
21-03-25 13:30:29.709	Ошибка выдачи реленой команды КПИ-Ф канал В
21-03-25 13:29:49.524	Ошибка инициализации алгоритмов КПИ-Ф канал А
21-03-25 13:29:49.524	Ошибка инициализации алгоритмов КПИ-Ф канал В
21-03-25 13:29:29.383	Ошибка инициализации АЦП КПИ-Ф канал А
21-03-25 13:29:29.383	Ошибка связи с Флеш накопителем 2 КПИ-Ф канал А
21-03-25 13:29:29.383	Ошибка инициализации АЦП КПИ-Ф канал В
21-03-25 13:29:29.383	Ошибка связи с Флеш накопителем 2 КПИ-Ф канал В

**Рисунок 8.8.1 – отображение блока «Таблица» на схеме в web-браузере**

Блок «Таблица» предназначен для вывода списка сработавших в настоящее время дискретных сигналов. Т.е. их значения равны 1. На объекте может быть много сигналов, но сработавших обычно немного. Блок сортирует сигналы в соответствии с настройкой. Для каждого приоритета можно задать цвет. Чем выше номер приоритета, тем он важнее.

Перечень свойств блока приведен в таблице 8.8.1:

№ п/п	Название свойства	Описание свойства
1.	Таблица с сигналами и их описанием	Набор строк в следующем формате: Имя сигнала;Описание;важность;номер в массиве;номер канала Пример: 701gtv001_state_valve;Ошибка выдачи реленой команды КПИ-Ф канал А;2;1;1 701gtv005_state_valve;Ошибка инициализации Ethernet КПИ-Ф канал А;2;1;1 701gtv003_state_valve;Ошибка инициализации SPI Флеш КПИ-Ф канал А;2;1;1 Если канал один и сигнал не массив, то их можно не задавать: 701gtv001_state_valve;Ошибка выдачи реленой команды КПИ-Ф;2
2.	Видимый цвет фона	Если «Да», то цвет фона блока можно задать в свойстве «Цвет фона».
3.	Цвет текста	Цвет текста
4.	Размер шрифта	Размер шрифта
5.	Цвет области заголовка	Цвет области заголовка
6.	Цвет линии таблицы	Цвет линии таблицы



7.	Цвет приоритета 0, 1, 2, 3	Цвета приоритетов.
8.	Дополнительные цвета приоритетов (начиная с 4)	Формат: Номер приоритета;цвет 4;#DC143C
9.	Сортировка	Варианты: - По времени. Строки сортируются по убыванию меток времени сработавших сигналов. Вверху оказываются самые свежие сигналы; - По важности, потом по времени. Строки сортируются сначала по важности в порядке убывания. Вверху оказываются самые важные сигналы. Внутри одной важности строки сортируются по времени.

## 8.9 CheckVox (галочка)



**Рисунок 8.9.1 – отображение блока «CheckVox» на схеме в web-браузере, когда значение сигнала равно 1**

Блок «CheckVox» предназначен для вывода и ввода значения дискретного сигнала. Если значение сигнала равно 0, то галочка внутри прямоугольника не отображается. Если значение сигнала равно 1, то галочка внутри прямоугольника отображается. При нажатии мышкой на блоке происходит запись инвертированного значения в сигнал. Если значение сигнала было равным 1, то запишется значение 0. Если значение сигнала было равным 0, то запишется значение 1. Значение сигнала на схеме может обновляться с задержкой. Поэтому повторные нажатия по блоку через время менее 1 секунды игнорируются.

Перечень свойств блока приведен в таблице 8.9.1:

№ п/п	Название свойства	Описание свойства
1.	Имя сигнала	Имя сигнала до нижнего подчеркивания.
2.	Имя хвоста сигнала	Имя сигнала после нижнего подчеркивания.
3.	Работа с разными каналами	Если «Да», то позволяет выводить информацию о различиях значений и статусов по разным каналам.
4.	Номер канала, из которого отображать данные	0 – объединять значения и статусы из разных каналов. Если задать номер канала, то будут отображаться данные только из этого канала.



5.	Цвет текста при расхождении значений каналов	Если значения по каналам отличаются, текст подписи окрашивается в этот цвет.
6.	Текст подписи	Текст подписи справа от квадратика с галочкой
7.	Шрифт подписи	Шрифт подписи справа от квадратика с галочкой
8.	Цвет галочки	Цвет галочки
9.	Цвет квадратика	Цвет квадратика вокруг галочки
10.	Цвет квадратика, когда недостоверность	Цвет квадратика вокруг галочки, когда недостоверность
11.	Цвет рамки плохого статуса	Если плохой статус, вокруг блока появляется рамка этого цвета и немного моргает
12.	Записывать значения при клике	Варианты выбора: - Нет – при нажатии мышкой на блок ничего не происходит; - Записать инвертированное значение – При нажатии мышкой на блоке происходит запись инвертированного значения в сигнал. Если значение сигнала было равным 1, то запишется значение 0. Если значение сигнала было равным 0, то запишется значение 1.

## 8.10 Переключатель



**Рисунок 8.10.1 – отображение блока «Переключатель» на схеме в web-браузере, когда значение сигнала равно 1**

Блок «Переключатель» предназначен для вывода и ввода значения дискретного сигнала. Если значение сигнала равно 0, то кружок отображается в левой части блока. Если значение сигнала равно 1, то кружок отображается в правой части блока.

При нажатии мышкой на блоке происходит запись инвертированного значения в сигнал. Если значение сигнала было равным 1, то запишется значение 0. Если значение сигнала было равным 0, то запишется значение 1. Значение сигнала на схеме может обновляться с задержкой. Поэтому повторные нажатия по блоку через время менее 1 секунды игнорируются.

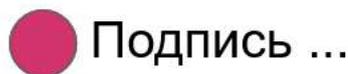


Перечень свойств блока приведен в таблице 8.10.1:

№ п/п	Название свойства	Описание свойства
1.	Имя сигнала	Имя сигнала до нижнего подчеркивания.
2.	Имя хвоста сигнала	Имя сигнала после нижнего подчеркивания.
3.	Работа с разными каналами	Если «Да», то позволяет выводить информацию о различиях значений и статусов по разным каналам.
4.	Номер канала, из которого отображать данные	0 – объединять значения и статусы из разных каналов. Если задать номер канала, то будут отображаться данные только из этого канала.
5.	Цвет текста при расхождении значений каналов	Если значения по каналам отличаются, текст подписи окрашивается в этот цвет.
6.	Текст подписи	Текст подписи справа от переключателя.
7.	Шрифт подписи	Шрифт подписи справа от переключателя.
8.	Цвет кружка "Включено"	Цвет кружка, который отображается в правой части блока, когда значение сигнала равно 1.
9.	Цвет линии кружка "Включено"	Цвет линии обводки кружка, который отображается в правой части блока, когда значение сигнала равно 1.
10.	Цвет кружка "Отключено"	Цвет кружка, который отображается в левой части блока, когда значение сигнала равно 0.
11.	Цвет линии кружка "Отключено"	Цвет линии обводки кружка, который отображается в левой части блока, когда значение сигнала равно 0.
12.	Цвет линии обводки (когда достоверно)	Цвет линии обводки кружка, когда сигнал достоверен.
13.	Цвет обводки и заливки кружка, когда недостоверность	Цвет обводки и заливки кружка, когда сигнал достоверен.
14.	Цвет рамки плохого статуса	Если плохой статус, вокруг блока появляется рамка этого цвета и немного моргает
15.	Записывать значения при клике	Варианты выбора: - Нет – при нажатии мышкой на блок ничего не происходит; - Записать инвертированное значение – При нажатии мышкой на блоке происходит запись инвертированного значения в сигнал. Если значение сигнала было равным 1, то запишется значение 0. Если значение сигнала было равным 0, то запишется значение 1.



## 8.11 Индикатор с возможностью управления



**Рисунок 8.11.1 – отображение блока «Индикатор с возможностью управления» на схеме в web-браузере**

Блок «Индикатор с возможностью управления» предназначен для вывода и ввода значения дискретного сигнала. Блок окрашивается разным цветом заливки в зависимости от значения сигнала, 0 или 1.

При нажатии мышкой на блоке и значении свойства «Записывать значения при клике» равным «Записать инвертированное значение» происходит запись инвертированного значения в сигнал. Если значение сигнала было равным 1, то запишется значение 0. Если значение сигнала было равным 0, то запишется значение 1. Значение сигнала на схеме может обновляться с задержкой. Поэтому повторные нажатия по блоку через время менее 1 секунды игнорируются.

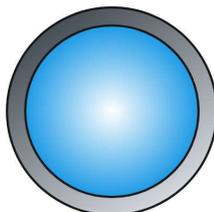
Перечень свойств блока приведен в таблице 8.11.1:

№ п/п	Название свойства	Описание свойства
1.	Имя сигнала	Имя сигнала до нижнего подчеркивания.
2.	Имя хвоста сигнала	Имя сигнала после нижнего подчеркивания.
3.	Работа с разными каналами	Если «Да», то позволяет выводить информацию о различиях значений и статусов по разным каналам.
4.	Номер канала, из которого отображать данные	0 – объединять значения и статусы из разных каналов. Если задать номер канала, то будут отображаться данные только из этого канала.
5.	Цвет текста при расхождении значений каналов	Если значения по каналам отличаются, текст подписи окрашивается в этот цвет.
7.	Текст подписи	Текст подписи справа от переключателя.
9.	Шрифт подписи	Шрифт подписи справа от переключателя.
10.	Цвет кружка "Включено"	Цвет кружка, который отображается в правой части блока, когда значение сигнала равно 1.
11.	Цвет кружка "Отключено"	Цвет кружка, который отображается в левой части блока, когда значение сигнала равно 0.
12.	Цвет линии обводки (когда достоверно)	Цвет линии обводки кружка, когда сигнал достоверен.
13.	Цвет обводки и заливки, когда недостоверность	Цвет обводки и заливки кружка, когда сигнал достоверен.
14.	Цвет рамки плохого статуса	Если плохой статус, вокруг блока появляется рамка этого цвета и немного моргает



15.	Записывать значения при клике	<p>Варианты выбора:</p> <ul style="list-style-type: none"> <li>- Нет – при нажатии мышкой на блок ничего не происходит;</li> <li>- Записать инвертированное значение – При нажатии мышкой на блоке происходит запись инвертированного значения в сигнал. Если значение сигнала было равным 1, то запишется значение 0. Если значение сигнала было равным 0, то запишется значение 1;</li> <li>- Нестандартное окно управления - при нажатии на блок появляется окно, созданное пользователем. Имя проекта окна задаётся в свойстве «Нестандартное окно управления».</li> <li>- Несколько нестандартных окон управления. Появляется окно с выбором из нескольких вариантов. Варианты нужно задать в свойстве «Нестандартные окна управления» в следующем формате:  <pre>sensor_h lim.prt;Ввод уставки Н LIM для датчика \$self.signal_name\$ sensor_hh lim.prt;Ввод уставки НН LIM для датчика \$self.signal_name\$</pre>           Сначала идет имя проекта окна управления. Затем идет строка, которую видит пользователь. Строка \$self.signal_name\$ заменяется на имя сигнала (до нижнего подчеркивания).</li> </ul>
-----	-------------------------------	--

## 8.12 Кнопка управления с автовозвратом



**Рисунок 8.12.1 – отображение блока «Кнопка управления с автовозвратом» на схеме в web-браузере**

Блок «Кнопка управления с автовозвратом» предназначен для вывода и ввода значения дискретного сигнала. Блок окрашивается разным цветом заливки в зависимости от значения сигнала, 0 или 1.

Особенностью блока является то, что при нажатии мышкой на блоке на десктопе или пальцем на сенсорном экране происходит запись в сигнал значения 1. При отпускании кнопки мышки происходит запись в сигнал значения 0. Запись значения возможна, если



значение сигнала перед нажатием равно 0. Если значение сигнала перед нажатием равно 1, выдаётся сообщение.

Перечень свойств блока приведен в таблице 8.12.1:

№ п/п	Название свойства	Описание свойства
1.	Имя сигнала	Имя сигнала до нижнего подчеркивания.
2.	Имя хвоста сигнала	Имя сигнала после нижнего подчеркивания.
3.	Работа с разными каналами	Если «Да», то позволяет выводить информацию о различиях значений и статусов по разным каналам.
4.	Номер канала, из которого отображать данные	0 – объединять значения и статусы из разных каналов. Если задать номер канала, то будут отображаться данные только из этого канала.
5.	Цвет фона при расхождении значений каналов	Если значения по каналам отличаются, фон окрашивается в этот цвет.
6.	Цвет текста при расхождении значений каналов	Если значения по каналам отличаются, текст окрашивается в этот цвет.
7.	Цвет "Включено"	Цвет, в который окрашивается фон блока, когда значение сигнала равно 1. В этот цвет окрашивается кнопка, когда пользователь давит на неё.
8.	Цвет "Отключено"	Цвет, в который окрашивается фон блока, когда значение сигнала равно 0.
9.	Цвет "Недостовверный"	Цвет, в который окрашивается фон блока, когда значение сигнала недостаточно. Например, когда нет связи между web-браузером и сервером.
10.	Менять цвет фона при плохом статусе	Если выбрано «Да», при плохом статусе цвет фона окрашивается в цвет, заданный в свойстве «Цвет фона при плохом статусе».
11.	Цвет рамки плохого статуса	Если плохой статус, вокруг блока появляется рамка этого цвета.
12.	Цвет линии	Цвет линии.
13.	Цвет обводки	Цвет рамки обводки вокруг кнопки.
14.	Дополнительный цвет градиента обводки	Дополнительный цвет градиента обводки рамки обводки вокруг кнопки.
15.	Цвет текста, если 0 (Отключено)	Цвет текста, введенного в свойстве «Текст, если 0» если значение сигнала равно 0.
16.	Цвет текста, если 1 (Включено)	Цвет текста, введенного в свойстве «Текст, если 1» если значение сигнала равно 1.
17.	Текст, если 0 (Отключено)	Текст, который отображается, если значение сигнала равно 0.
18.	Текст, если 1 (Включено)	Текст, который отображается, если значение сигнала равно 1.



### 8.13 Круглая кнопка



Рисунок 8.13.1 – отображение блока «Круглая кнопка» на схеме в web-браузере

Блок «Круглая кнопка» предназначен для записи значения сигнала. Блок ничего не отображает. При нажатии мышкой на блоке происходит запись значения в сигнал.

Перечень свойств блока приведен в таблице 8.13.1:

№ п/п	Название свойства	Описание свойства
1.	Имя сигнала	Имя сигнала, куда происходит запись, до нижнего подчеркивания.
2.	Имя хвоста сигнала	Имя сигнала, куда происходит запись, после нижнего подчеркивания.
3.	Значение для записи	Значение, которое записывается в сигнал, при нажатии на кнопку.
4.	Текст подписи	Текст подписи справа от кнопки.
5.	Шрифт подписи	Шрифт подписи справа от кнопки.
6.	Цвет заливки	Цвет заливки круга.
7.	Цвет линии	Цвет линии круга.

### 8.14 Ползунок

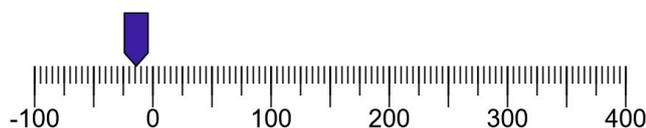


Рисунок 8.14.1 – отображение блока «Ползунок» на схеме в web-браузере

Блок «Ползунок» предназначен для вывода и ввода значения аналогового сигнала с окна схемы.

**В модальном окне не работает ввод данных при помощи ползунка! В модальном окне можно только посмотреть данные.**

**Ввод данных работает только при горизонтальном расположении ползунка.**



Положение бегунка на шкале показывает текущее значение сигнала. Если значение сигнала меньше минимума шкалы или больше максимума, бегунок будет мигать.

При нажатии мышкой на бегунке ползунка и последующем перемещении указателя мыши с нажатой клавишей бегунок будет следовать следом за указателем мыши. При отпускании кнопки мышки происходит запись в сигнал значения, на которое указывает бегунок относительно шкалы. При нажатии мышкой в области, где может перемещаться бегунок, происходит запись в сигнал значения шкалы в этом месте.

Перечень свойств блока приведен в таблице 8.14.1:

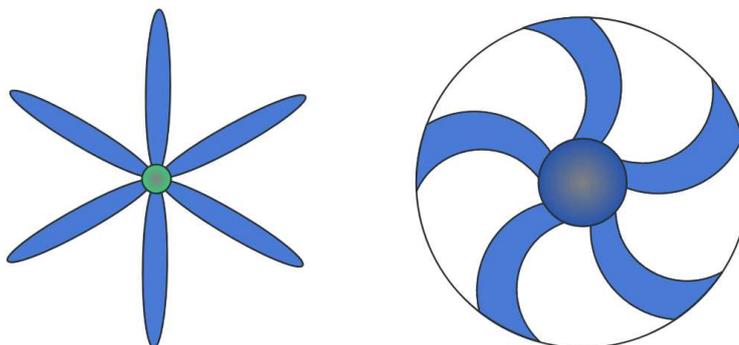
№ п/п	Название свойства	Описание свойства
1.	Имя сигнала	Имя сигнала до нижнего подчеркивания.
2.	Имя хвоста сигнала	Имя сигнала после нижнего подчеркивания.
3.	Работа с разными каналами	Если «Да», то позволяет выводить информацию о различиях значений и статусов по разным каналам.
4.	Номер канала, из которого отображать данные	0 – объединять значения и статусы из разных каналов. Если задать номер канала, то будут отображаться данные только из этого канала.
5.	Цвет бегунка при расхождении значений каналов	Если значения по каналам отличаются, фон бегунка окрашивается в этот цвет.
6.	Допустимая величина расхождения значений каналов	Если значения по каналам отличаются более, чем на эту величину, фон бегунка окрашивается в цвет, заданный в свойстве «Цвет бегунка при расхождении значений каналов».
7.	Менять цвет фона при плохом статусе	Если выбрано «Да», при плохом статусе цвет фона окрашивается в цвет, заданный в свойстве «Цвет фона при плохом статусе».
8.	Цвет рамки плохого статуса	Если плохой статус, вокруг блока появляется рамка этого цвета.
9.	Цвет бегунка ползунка	Цвет, в который окрашивается фон бегунка, когда значение сигнала достоверно и в норме.
10.	Цвет линии бегунка ползунка	Цвет линии бегунка ползунка.
11.	Индикация выхода за допустимый диапазон	Если поставить эту галочку, то можно задать диапазоны значений, при выходе за которые будет меняться цвет фона бегунка.
12.	Источник информации о выходе за уставку	Варианты: - Свойства. Для индикации значения аварийного и предупредительного берутся из свойств блока; - Другой сигнал. Для индикации значения информация о выходе за уставку берется из сигнала, заданном в свойстве «Имя хвоста сигнала выхода за уставку». Значения этого сигнала: 0 – норма, 1 – предупредительное, 2 – аварийное.



13.	Верхнее аварийное значение, Верхнее предупредительное значение, Нижнее предупредительное значение, Нижнее аварийное значение	Значения, при выходе за которые будет меняться цвет фона бегунка.
14.	Цвет бегунка аварийного значения	Цвет фона бегунка, который будет, если значение меньше нижнего аварийного значения или значение больше верхнего аварийного значения
17.	Цвет бегунка предупредительного значения	Цвет фона бегунка, который будет, если значение меньше нижнего предупредительного значения или значение больше верхнего предупредительного значения
18.	Нижний предел	Начальное значение шкалы.
19.	Верхний предел	Максимальное значение шкалы.
20.	Шаг	Размер промежутка в единицах шкалы для отображения цифровых подписей над рисками шкалы.
21.	Число делений на шаг	Число отображаемых рисок шкалы между соседними цифровыми подписями, включая риски с подписями.
22.	Высота штриха (% от размеров)	Высота штриха (% от размеров).
23.	Высота деления (% от размеров)	Высота деления (% от размеров).
24.	Показывать деления	Показывать деления.
25.	Показывать числа	Показывать числа.
26.	Шрифт	Шрифт цифровых подписей шкалы.
27.	Число больших делений на шаг	Число больших делений на шаг.
28.	Толщина малого штриха	Толщина малого штриха.
29.	Толщина большого штриха	Толщина большого штриха.
30.	Высота среднего штриха (% от размеров)	Высота среднего штриха (% от размеров)
31.	Количество средних штрихов на шаг	Количество средних штрихов на шаг



## 8.15 Вращающиеся элементы



**Рисунок 8.15.1 – отображение блоков «Вращающихся элементов» на схеме в web-браузере**

Блоки «Вращающийся элемент» и «Вращающийся элемент насоса» предназначены для индикации работы какого-либо оборудования. Когда оборудование работает, происходит индикация вращением.

Перечень свойств блока приведен в таблице 8.15.1:

№ п/п	Название свойства	Описание свойства
1.	Имя сигнала необходимости вращения	Имя сигнала до нижнего подчеркивания. Когда значение сигнала равно 1, блок вращается. В других случаях нет.
2.	Имя хвоста сигнала необходимости вращения	Имя сигнала после нижнего подчеркивания.
3.	Работа с разными каналами	Если «Да», то позволяет выводить информацию о различиях значений и статусов по разным каналам.
4.	Номер канала, из которого отображать данные	0 – объединять значения и статусы из разных каналов. Если задать номер канала, то будут отображаться данные только из этого канала.
5.	Цвет текста при расхождении значений каналов	Если значения по каналам отличаются, текст окрашивается в этот цвет.
6.	Цвет лопастей	Цвет лопастей блока.
7.	Цвет круга посередине	Цвет круга посередине
8.	Цвет линии лопастей и круга посередине	Цвет линии лопастей и круга посередине
9.	Текст подписи	Текст подписи справа от блока.
10.	Шрифт подписи	Шрифт подписи справа от блока.
11.	Показать обводку	Показать круг вокруг блока
12.	Цвет линии обводки (когда достоверно)	Цвет линии обводки (когда достоверно)
13.	Цвет обводки и заливки, когда недостоверность	Цвет обводки и заливки, когда недостоверность



14.	Цвет рамки плохого статуса	Если плохой статус, вокруг блока появляется рамка этого цвета.
17.	Источник скорости вращения	Варианты: - Из свойства "Скорость вращения". Скорость вращения задаётся в свойстве «Скорость вращения на схеме (об/мин)» и измеряется в оборотах в минуту. Рекомендуется задавать не более 60, иначе высокая скорость будет привлекать излишнее внимание. - Из сигнала. Скорость вращения берется из сигнала, название которого оканчивается на значение свойства «Хвост сигнала скорости вращения (об/мин)». Если скорость вращения отрицательная, лопасти будут вращаться в другую сторону.
18.	Направление вращения	Можно выбрать направление вращения «По часовой стрелке» или «Против часовой стрелки»

## 8.16 Воспроизведение звука

Блок «Воспроизведение звука» предназначен для воспроизведения звука, когда значение какого-то из заданных сигналов равно заданному значению. Звук воспроизводится непрерывно пока значение сигнала равно заданному.

Блок не должен быть виден оператору, т.к. он не отображает никаких значений. Поэтому рекомендуется помещать его на схеме вне области видимости. Если вы хотите, чтобы блок действовал на всех схемах, поместите его на верхнее или нижнее меню.

Список сигналов, значения которых нужно контролировать задается в свойстве «Имена сигналов (столбиком)».

Пример записи списка сигналов:

КВА00AA3\_state

КВА00AA5\_state

КВА00AA3\_in

КВА00AA5\_in

КВА00AA3\_ina

КВА00AA5\_ina

Имена звуковых файлов и значения, при которых они должны звучать, задаются в свойстве «Имена звуковых файлов;Значения». В каждой строке сначала записывается имя звукового файла, затем точка с запятой, затем значение сигнала, при котором будет звучать



этот файл. Если значение хотя бы одного из сигналов, заданных в свойстве «Имена сигналов (столбиком)», равно заданному в строке, то будет звучать этот файл.

Одновременно не могут звучать несколько звуковых файлов. Если у разных сигналов разные значения, то приоритет отдается тому файлу, который выше в списке.

Пример записи списка звуковых файлов и значений:

/sounds/fantastic-siren.mp3;3

/sounds/groundhog-squeak-warning-of-danger.mp3;2

/sounds/the-bewildered-meow-of-a-grown-up-e-kid.mp3;1

Для символа разделителя должен использоваться /

Физически звуковые файлы должны находиться в папке web-сервера. В нашем примере в корневой папке web-сервера находится папка sounds, в которой располагаются файлы. Папку sounds можно скопировать вручную на web-сервер.

По умолчанию в браузере запрещено воспроизводить звуки на web-странице. Чтобы работал блок, нужно внести изменения в настройки браузера.

В браузере Google Chrome в правом верхнем углу окна нажать на значок с тремя точками, затем «Настройки». Выбрать «Конфиденциальность и безопасность», далее «Настройки сайтов». В разделе «Контент» нажать «Дополнительные настройки контента», «Звук». В разделе «Настройки по умолчанию» выбрать опцию «Разрешить сайтам воспроизводить звук». В разделе «Специальные настройки» найти подраздел «Разрешить сайтам воспроизводить звук», нажать кнопку «Добавить», ввести адрес web-сервера. Например, <http://127.0.0.1:8088>

В браузере Firefox зайти в настройки (about:preferences). Выбрать панель «Приватность и защита» и перейти вниз к разделу «Разрешения». Щёлкнуть по кнопке «Параметры...» рядом с «Автовоспроизведение». Использовать выпадающее меню «По умолчанию для всех сайтов», выбрать «Разрешить аудио и видео». Так же можно предоставить разрешения для сайта непосредственно из кнопки, появляющейся в адресной строке.



**Рисунок 8.15.1 – кнопка предоставления разрешений для сайта в web-браузере Firefox**



## 8.17 Блоки теплогидравлики

Блоками являются задвижка, обратный клапан, насос, электродвигатель, бак. Свойства этих блоков аналогичны свойствам предыдущих блоков. В группе блоков «Оформление» можно задать цвета и тип оборудования.

Блок «Электродвигатель или генератор» может отображаться изображением либо электродвигателя, либо генератора. Тип отображения можно выбрать в группе «Оформление» в свойстве «Тип графического примитива».

Задвижка, обратный клапан отображают значение сигнала положения. Значение сигнала должно быть в диапазоне от 0 до 100.

Насос, электродвигатель отображают значение сигнала состояния. Значение сигнала должно быть 0 или 1.

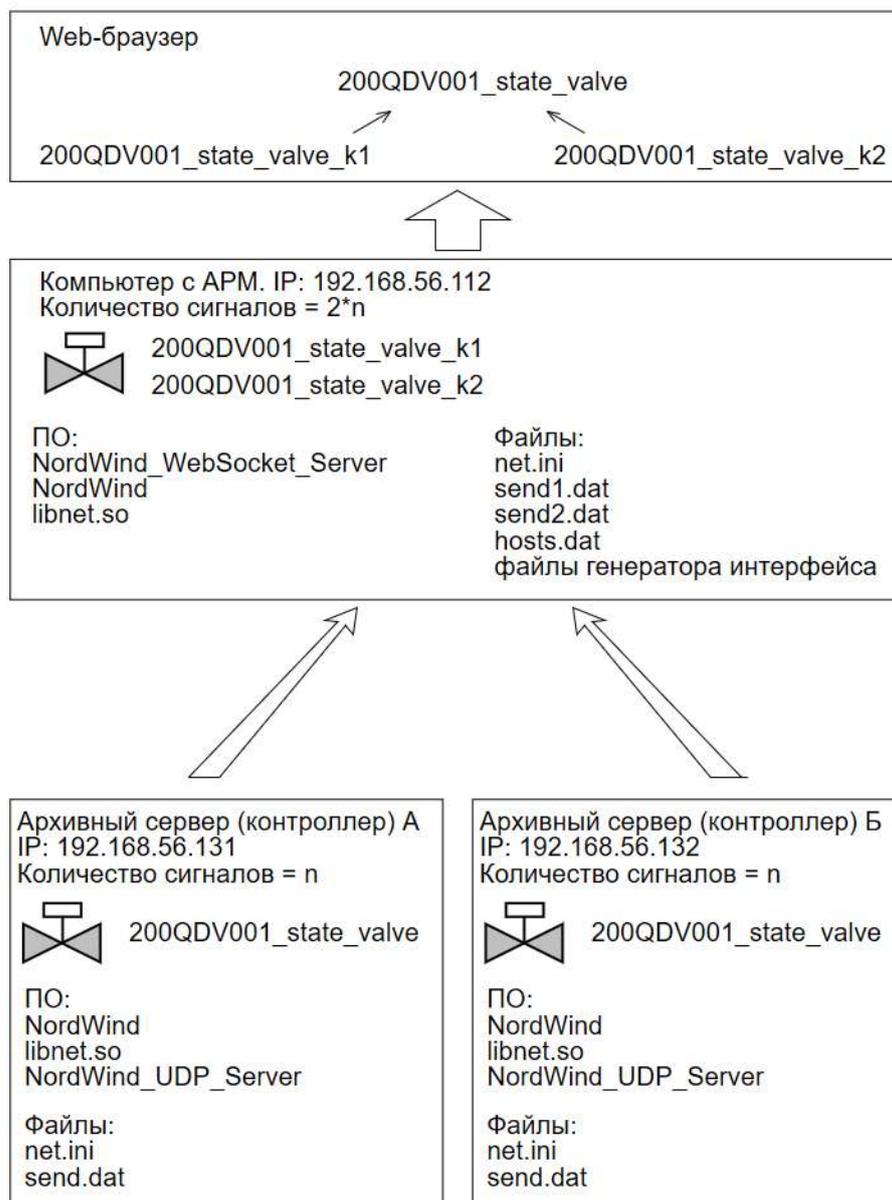
Бак отображает значение уровня. Минимальное и максимальное значения уровней должно быть задано в свойствах «Нижний предел», «Верхний предел».



## 9 Работа с несколькими каналами

### 9.1 Архитектура системы

Архитектура системы при работе с несколькими каналами представлена на рисунке:



**Рисунок 9.1 – работа с несколькими каналами**

Есть два архивных сервера (контроллера) с одинаковым алгоритмом работы и одинаковым набором сигналов. Для объединения данных от двух серверов и отображения информации используется третий компьютер с NordWind и web-браузером. Этот NordWind имеет алгоритм, который принимает набор данных с каждого контроллера. Он имеет базу данных, состоящую из двойного списка сигналов. При этом к имени сигнала от первого



контроллера добавляется строка `_k1`, к имени сигнала от второго контроллера добавляется строка `_k2`. В браузер передается удвоенный список сигналов. В браузере сигналы с `_k1` и `_k2` на концах объединяются в один. И отображаются пользователю как один. При создании проекта для генератора интерфейса имена сигналов задаются без `_k1` и `_k2`.

Единственное отличие проекта, работающего с двумя каналами от обычного, является настройка «Количество каналов обмена». Её нужно установить в 2.

Для контроля исправности канала связи в проект, исполняемый на архивном сервере (контроллере), можно добавить сигнал с именем `connected_signal`. Значение этого сигнала должно постоянно меняться.

Для визуализации исправности канала на схему следует добавить блок «ИНТЕРФЕЙС – Дискретный». В свойстве «Имя сигнала» нужно написать `connectedsignal`. Свойство «Имя хвоста сигнала» должно быть пустое. Поставить галку «Работа с разными каналами». В свойстве «Номер канала, из которого отображать данные (0 - объединять)» нужно ввести номер канала, исправность которого нужно визуализировать (нумерация с 1).

Если значение сигнала с именем `connected_signal` не меняется дольше двух секунд, то канал обмена считается неисправным, значение `connectedsignal` будет равным 0. Если значение сигнала с именем `connected_signal` меняется часто, то канал обмена считается исправным, и значение `connectedsignal` будет равным 1.

## **9.2 ПО и конфигурационные файлы, устанавливаемые на контроллерах**

Рассмотрим ПО и конфигурационные файлы, устанавливаемые на контроллерах.

На каждом контроллере устанавливается ПО:

- NordWind;
- `libnet.so`. Для пересылки данных от контроллера в компьютер с АРМ;
- `NordWind_UDP_Server`. Для приема команды записи значения от АРМ (`NordWind_WebSocket_Server`) и записи его в базу данных NordWind.

Конфигурационные файлы, размещаемые на контроллерах:

- `net.ini`. Задается хост, порт компьютера с АРМ, на который отправляются данные. И имя файла `send.dat`;
- `send.dat`. Содержит список сигналов, которые отправляются на компьютер с АРМ (без `_k1` и `_k2`);



Пример файла net.ini для контроллера А:

```
[Send0]
port=23577
host=192.168.56.112
file=send.dat
takt=10
```

Пример файла net.ini для контроллера Б:

```
[Send1]
port=23578
host=192.168.56.112
file=send.dat
takt=10
```

### **9.3 ПО и конфигурационные файлы, устанавливаемые на компьютере с АРМ**

На компьютере с АРМ устанавливается ПО:

- NordWind;
- libnet.so. Для приема данных от контроллеров А и Б;
- NordWind\_WebSocket\_Server. Является web-сервером;
- arm.so. Библиотека с алгоритмом с полным списком сигналов. Кодогенерируется из проекта MergeForARM.prt.
- Web-браузер для отображения информации оператору. Также может размещаться и на других компьютерах в сети.

Конфигурационные файлы, размещаемые на компьютере с АРМ:

- net.ini. Задаются порты, по которым принимаются данные от каждого контроллера, и имена файлов сигналов (send1.dat и send2.dat), которые принимаются от каждого контроллера (с \_k1 или \_k2 на конце);
- send1.dat. Содержит список сигналов, которые принимаются от контроллера А (с \_k1 на конце);
- send2.dat. Содержит список сигналов, которые принимаются от контроллера Б (с \_k2 на конце);
- hosts.dat. Содержит список IP-адресов контроллеров А и Б. Нужен для NordWind\_WebSocket\_Server, чтобы он знал, куда отправлять команды на запись значений.



Пример файла net.ini:

```
[Receive0]
port=23577
file=send1.dat
timeout=1000

[Receive1]
port=23578
file=send2.dat
timeout=1000
```

Пример файла hosts.dat:

```
192.168.56.131
192.168.56.132
```

## **9.4 Генерация проекта, который будет работать на компьютере с АРМ**

Для генерации проекта, который будет работать на компьютере с АРМ необходим список сигналов, находящихся на контроллере. Для получения этого списка при кодогенерации проекта, исполняемого на контроллере в окне кодогенератора Си в вкладке «Настройки» необходимо поставить галочку «Генерировать вспомогательные списки входов и выходов». Далее необходимо выполнить генерацию кода. После генерации в директории исходников появятся два файла с расширениями `inputs` и `outputs`. В окне генератора интерфейса на вкладке «настройка» поле «Количество каналов обмена» должно содержать 2. Должно быть видно поле ввода «Директория файлов `inputs` и `outputs`». В этом поле нужно задать директорию исходников, в которой появились два файла с расширениями `inputs` и `outputs`. Можно вводить шаблон. Подстрока `%currentdir%` при генерации заменяется на полный путь к файлу конфигурации `vk`. Например, если создать в папке с конфигурацией папку `src_linux`, то в это поле можно написать `%currentdir%\src_linux\`. Далее на вкладке «Загрузка» необходимо нажать на кнопку «Создать проект для АРМ» . После этого в паке, в которой находится конфигурация `vk` появится папка `GenerateForMergeServerOnARM`. В ней появится проект `MergeForARM.prt` для компьютера с АРМ. Далее произойдет кодогенерация этого проекта. Настройки кодогенератора берутся из окна «Кодогенератор Си». Должны появиться файлы `arm.so` и



другие для загрузки на компьютер с АРМ. Директория, где появятся эти файлы берется из окна «Кодогенератор Си». Так же в папке GenerateForMergeServerOnARM появятся файлы net.ini, send1.dat, send2.dat. Их нужно скопировать на компьютер с АРМ. На компьютере с АРМ необходимо создать файл hosts.dat со списком IP-адресов контроллеров А и Б.

Появятся папки \_k1 и \_k2. Файлы net.ini и send.dat из папок \_k1 (\_k2) необходимо скопировать на контроллер А (Б). Файл net.ini создается для примера, в нем необходимо исправить параметр host. Его достаточно скопировать один раз.



## 10 Создание новых блоков

### 10.1 Основные положения

Незначительные изменения в блоках требуют от программиста обязательных начальных знаний языка JavaScript, SVG и умения пользоваться инструментами разработчика в браузере. В данном документе они не рассматриваются.

В процессе генерации для каждой схемы создаются файл html и по 3 выходных файла в папке js-generate.

init-Имя\_схемы.js — файл с функциями, которые выполняются сразу же после загрузки web-страницы.

vars-Имя\_схемы.js — файл с переменными и объектами. Каждый блок представлен объектом.

step-Имя\_схемы.js — файл с функциями, которые выполняются при каждом изменении сигнала, входящего в блок. Обычно функция занимается изменением внешнего вида блока на схеме.

JavaScript функции записываются в соответствующих свойствах блоков. При генерации текстовое содержимое этих свойств выводится в выходные файлы в нужной последовательности.

В блоке могут быть следующие свойства:

Скрипт инициализации (script\_init) — выполняется сразу же после загрузки web-страницы.

Скрипт на шаге (script\_step) — выполняется при изменении значения или статуса сигнала.

Скрипт onclick (script\_onclick) — выполняется при клике мышкой по блоку.

Скрипт mouseover (script\_mouseover) — выполняется, когда курсор мыши над блоком.

Скрипт mouseout (script\_mouseout) — выполняется, когда курсор мыши уходит от блока.

Скрипт mousedown (script\_mousedown) — выполняется, когда пользователь нажимает клавишу мыши на блоке или наживает пальцем на блоке на устройстве сенсорного ввода (смартфон, планшет).



Скрипт `mouseup (script_mouseup)` — выполняется, когда пользователь отпускает клавишу мыши на блоке или отпускает палец с блока на устройстве сенсорного ввода (смартфон, планшет).

Содержимое свойств со скриптами может быть пустым. Свойства находятся в группе `JavaScripts`. Если они не видны, значит они скрыты. Для появления их нужно на вкладке общие, в свойстве «Список невидимых свойств (`InvisibleProps`)» стереть строки `script_init; script_step; script_onclick; script_mouseover; script_mouseout`.

Внутри скрипта мы можем получить значение свойства блока через запись вида: `$self.prefix$`. Где `prefix` — это имя свойства. При генерации файлов происходит замена в тексте этой записи на значение свойства. Если свойство имеет тип данных «двоичное», то выходное значение будет `true` или `false`. Если свойство имеет тип данных «цвет», то выходное значение будет представлено в виде цвета `html`, например `#F9FE40`. Стоит помнить, что в `JavaScript` цвет, как правило задается в виде строки, нужны кавычки. Поэтому писать надо, например так: `let lo_level_background_color = "$self.color$";`

Внутри скрипта мы можем получить значение сигнала из базы данных и значения свойств из параметров расчёта окна «параметры проекта» через запись вида: `$trendclient_port$`. Где, `trendclient_port` – это имя свойства или сигнала БД.

Узнать текущее значение в скрипте можно через запись `aObj[signName]`. `signName` — это имя сигнала. Имя сигнала должно быть только маленькими буквами. В скрипте `script_init` невозможно узнать текущее значение сигнала, т. к. он выполняется раньше, чем будут получены значения. Если сигнала нет в сервере или произошел разрыв соединения между сервером и браузером, в `aObj[signName]` будет `undefined`.

Узнать текущий статус сигнала в скрипте можно следующим образом:

```
let nameStatus = getStatusName(signName, 0);
```

```
let status = aObj[nameStatus];
```

Если количество каналов обмена больше 1, то получить имя можно следующим образом:

```
nameStatus = getStatusName(signName, numberChannel); // где numberChannel – номер канала. Начинается с 1.
```

Новый сигнал нужно добавить функцией `dbgaddreadvar`. Это нужно сделать в скрипте «Скрипт инициализации (`script_init`)». Пример:

```
dbgaddreadvar(aObj,"my_signal",aObj.name);
```



Первый аргумент у функции всегда `aObj. "my_signal"` — имя сигнала, следует заранее позаботиться, чтобы он был маленькими буквами.

Перед написанием скрипта полезно посмотреть на сгенерированный html-файл.

В html-файле содержатся svg-примитивы.

Пример svg-примитива:

```
<rect id="Analog.FillRect" x="-24" y="-8" width="48" height="16" stroke-width="1" stroke="rgb(255,255,255)" fill="rgb(59,20,175)" style="opacity: 1"/>
```

У каждого svg-примитива есть свой `id`. Он состоит из имени блока, точки и имени примитива. `id` текста немного другой. К концу добавляется `_line0`. Например, `Analog.TextVal_line0`.

Установить новый атрибут у примитива блока можно функцией **SetAttr**. Она описана в файле `videokadr.js`.

Пример: `SetAttr(aObj, "FillRect", "fill", "#F9FE40");`

Первый аргумент у функции всегда `aObj. "FillRect"` — имя графического примитива, при генерации спереди к нему добавляется имя блока с точкой. `"fill"` — имя атрибута у сгенерированного svg-примитива, в нашем случае цвет заливки. `"#F9FE40"` — новое значение атрибута, то есть цвет.

Установить новый текст можно следующим образом:

`SetAttr(aObj, "TextVal_line0", "innerHTML", "новый текст");`

Функция **floatToStrFormat** форматирует число с плавающей запятой. `floatToStrFormat(value, num_precision, num_digits, num_format)`

`value` – значение;

`num_precision` – количество знаков общее;

`num_digits` – количество знаков после запятой;

`num_format` – формат числа.

Узнать значение атрибута у примитива блока можно функцией **GetAttr**.

Пример: `strokeWidth = GetAttr(aObj, "FillCircle", "stroke-width");`

Первый аргумент у функции всегда `aObj. "FillCircle"` — имя графического примитива, `"stroke-width"` — имя атрибута у сгенерированного svg-примитива, в нашем случае толщина линии. Значение толщины линии будет в переменной `strokeWidth`.

Записать новое значение и статус в сигнал можно функцией **setval**.

Пример: `setval("kba00aa5_in", "1", "0");`



Первый аргумент у функции имя сигнала. Второй аргумент новое значение, в нашем случае 1. Третий аргумент новый статус, в нашем случае пусть будет 0, т.к. это означает, что всё в норме.

Можно записать только значение или только статус. Для этого есть функция **setOnlyValOrStatus**.

Пример: `setOnlyValOrStatus("kba00aa5_in", "1", writeValue)`; Первый аргумент у функции имя сигнала. Второй аргумент новое значение, в нашем случае 1. Третий аргумент константа `writeValue` указывает, что нужно записать только значение. Для записи одного статуса следует написать `setOnlyValOrStatus("kba00aa5_in", "1", writeStatus)`; Статус будет равен 1.

Для показа пользователю диалогового окна с кнопкой ОК предназначена функция **showAlert**.

Пример: `showAlert("Значение сигнала недостоверно", "Ошибка")`; выведет на экран диалог, представленный на рисунке.

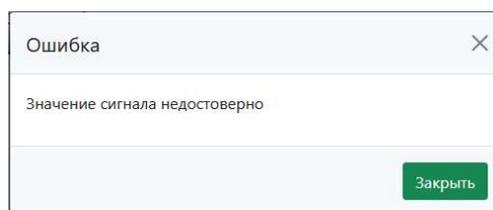


Рисунок 10.1.1 – диалог в web-браузере

## 10.2 Простой пример блока для вывода значения сигнала

Рассмотрим пример блока «ИНТЕРФЕЙС - Аналоговый без фона (учебный)» со вкладки «Учебные».

Откроем значение свойства «Скрипт инициализации (`script_init`)». Этот скрипт запускается один раз при загрузке web-страницы.

Сначала вычислим полное имя сигнала. Для этого сложим начало имени сигнала, символ нижнего подчеркивания и хвост сигнала:

```
// Получаем имя сигнала до хвоста
let signName = aObj.signalName;

// Присоединяем хвост к имени сигнала
let signPostfix = "$self.postfix$".toLowerCase();
```



```
if (signPostfix != ""){  
    aObj.FullSignalName = signName + "_" + signPostfix;  
}
```

Теперь нам нужно указать серверу, что мы хотим получать этот сигнал от него. Это нужно сделать функцией `dbgaddreadvar`:

```
dbgaddreadvar(aObj, aObj.FullSignalName, aObj.name);
```

После загрузки web-страницы при изменении значения или статуса сигнала будет срабатывать скрипт «Скрипт на шаге (`script_step`)». Откроем значение этого свойства.

В нем нам нужно узнать значение сигнала и установить это значение в нужный атрибут блока.

```
// Узнаем значение сигнала
```

```
let value = aObj[aObj.FullSignalName];
```

```
// Приводим число к строке. 8 цифр перед запятой, 2 цифры после запятой, 2 -  
фиксированный формат
```

```
let valueFormat = floatToStrFormat(value, 8, 2, 2);
```

```
// Устанавливаем HTML-содержимое элемента (т.е. выводим новый текст)
```

```
SetAttr(aObj, "TextVal_line0", "innerHTML", valueFormat);
```

### **10.3 Получение свойств блока в скрипте. Изменение фона прямоугольника по условию**

Задача добавить фон в предыдущий блок и менять его цвет по условию.

Зайдем в графический редактор блока «ИНТЕРФЕЙС - Аналоговый без фона (учебный)» со вкладки «Учебные». Добавим графический примитив «залитый прямоугольник». Свойство «Имя объекта» установим `FillRect`.

Добавим новые свойства в блок через правую кнопку мыши – «изменить блок...». Добавим свойство «Верхняя уставка» с именем `hi_level`.

Внутри скрипта мы можем получить значение свойства блока через запись вида: `$self.hi_level$`. Где `hi_level` — это имя свойства. При генерации файлов происходит замена в тексте этой записи на значение свойства.

Откроем значение свойства «Скрипт на шаге (`script_step`)», и допишем там следующее:

```
// Определяем переменные со значениями свойств
```



```

let hiLevel = $self.hi_level$; // Верхняя уставка
let loLevel = $self.lo_level$; // Нижняя уставка
let sensorColor = "$self.sensor_color$"; //Цвет в виде строки, поэтому заключаем
значение в кавычки
let levelColor = "$self.level_color$"; //Цвет в виде строки, поэтому заключаем
значение в кавычки
Далее напишем условие, в котором мы будем менять цвет прямоугольника в
зависимости от значения сигнала:
if ((value > hiLevel) || (value < loLevel)){//Значение выходит за уставку
    // Изменяем значение цвета заливки fill у прямоугольника с именем FillRect
    SetAttr(aObj, "FillRect", "fill", levelColor);
} else{//Значение в норме
    SetAttr(aObj, "FillRect", "fill", sensorColor);
}

```

#### 10.4 Работа со статусом сигнала

Задача изменять цвет линии рамки фона в зависимости от статуса сигнала.

Добавим в блок следующие свойства:

- Цвет линии рамки при хорошем статусе (line\_color)
- Цвет линии рамки при плохом статусе (line\_bad\_color)

Внутри скрипта на шаге мы получим значения свойств:

```

let lineColor = "$self.line_color$"; // Определим значение цвета линии рамки при
хорошем статусе
let lineBadColor = "$self.line_bad_color$"; // Определим значение цвета линии рамки
при плохом статусе
// Узнаем значение статуса сигнала
let nameStatus = getStatusName(aObj.FullSignalName, 0);
let status = aObj[nameStatus];
if (status === 0){
    // Изменяя атрибут с именем stroke мы изменяем цвет линии прямоугольника
    SetAttr(aObj, "FillRect", "stroke", lineColor);
} else {
    SetAttr(aObj, "FillRect", "stroke", lineBadColor);
}

```

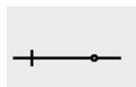


}

## 10.5 Работа с видимостью графических примитивов

Задача создать блок, отображающий положение рубильника (выключателя).

Рассмотрим пример блока «ИНТЕРФЕЙС - Дискретный (учебный)» со вкладки «Учебные».



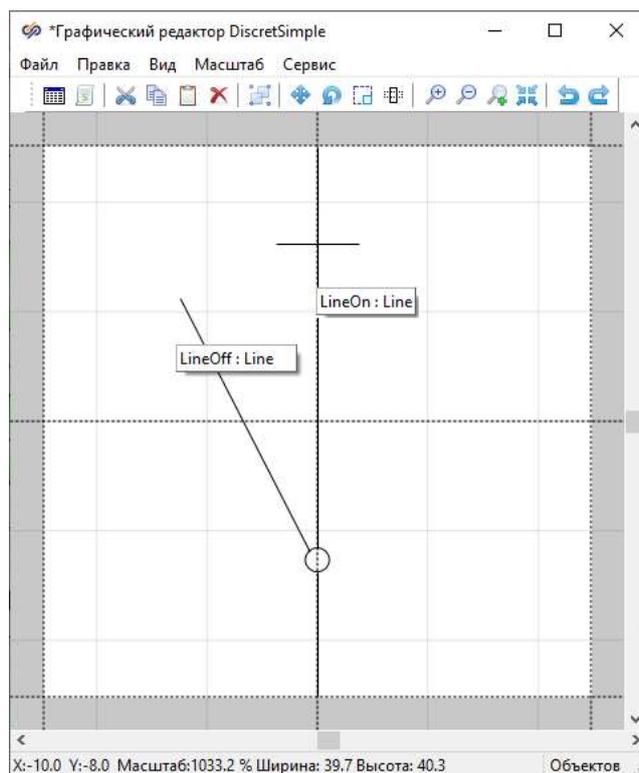
- Внешний вид блока во включённом состоянии.



- Внешний вид блока в отключённом состоянии.

Значение свойства «Скрипт инициализации (script\_init)» аналогично предыдущему блоку.

Через «графическое изображение блока...» откройте графический редактор, посмотрите изображение.



У блока есть 2 линии, видимость которых будет меняться в зависимости от состояния. LineOn – будет видна, когда он включен. LineOff – будет видна, когда он отключен. Видимостью этих линий будем управлять из скрипта следующим образом:

```
SetAttr(aObj, "LineOn", "style", "opacity: 1"); // Делаем линию видимой
```

```
SetAttr(aObj, "LineOff", "style", "opacity: 0"); // Делаем линию невидимой
```



Откроем значение свойства «Скрипт на шаге (script\_step)», и допишем там следующее:

```
if (value === 1){//Отображаем положение включено
  SetAttr(aObj, "LineOn", "style", "opacity: 1"); // Делаем линию видимой
  SetAttr(aObj, "LineOff", "style", "opacity: 0"); // Делаем линию невидимой
} else
  {//Отображаем положение отключено. Эта часть условия выполнится так же если
  значение сигнала неопределено
  SetAttr(aObj, "LineOn", "style", "opacity: 0"); // Делаем линию невидимой
  SetAttr(aObj, "LineOff", "style", "opacity: 1"); // Делаем линию видимой
}
```

## 10.6 Запись значений

Задача создать блок, записывающий новое значение при клике на нем мышкой.

Рассмотрим пример блока «ИНТЕРФЕЙС - Круглая кнопка (учебный)» со вкладки «Учебные».

Откроем значение свойства «Скрипт инициализации (script\_init)». Этот скрипт запускается один раз при загрузке web-страницы.

Сначала вычислим полное имя сигнала. Для этого сложим начало имени сигнала, символ нижнего подчеркивания и хвост сигнала:

```
// Получаем имя сигнала до хвоста
let signName = aObj.signalName;
aObj.FullSignalName = signName;
// Присоединяем хвост к имени сигнала
let signPostfix = "$self.postfix$".toLowerCase();
if (signPostfix !== ""){
  aObj.FullSignalName = signName + "_" + signPostfix;
}
```

Никаких сигналов мы не хотим отображать на блоке, поэтому больше ничего не пишем.

Запись значения должна выполняться при клике мышкой по блоку. Поэтому откроем значение свойства «Скрипт onclick (script\_onclick)»

Напишем в нем функцию, которая записывает новое значение и статус в сигнал:

```
setval(aObj.FullSignalName, "$self.value_write$", "0");
```



Первый аргумент у функции имя сигнала. Второй аргумент новое значение. Новое значение берем из окна редактора свойств, свойства `value_write` с помощью конструкции `"$self.value_write$"`. Третий аргумент новый статус.

Рассмотрение более сложных примеров будет требовать от программиста знаний языка JavaScript на среднем уровне. И не имеет смысла по причине того, что, по сути, будет пересказом элементов языка JavaScript.



## 11 Полезные настройки браузера

Чтобы запретить переход вкладок Firefox в спящий режим, нужно отключить функцию `browser.tabs.unloadOnLowMemory`. Для этого необходимо выполнить следующие шаги:

- Открыть браузер Firefox;
- Перейти на страницу дополнительных настроек. Для этого в адресной строке ввести `about:config` и нажать клавишу `Enter`;
- Появится предупреждающее сообщение «Действовать с осторожностью». Нужно нажать на кнопку «Принять риск и продолжить», откроется страница дополнительных настроек;
- В поле поиска ввести `browser.tabs.unloadOnLowMemory`;
- Когда настройка будет видна, дважды щёлкнуть по ней, чтобы установить значение `false`.

Чтобы запретить переход вкладок в спящий режим в Chrome, нужно отключить функцию «Экономия памяти». По умолчанию она включена и временно приостанавливает неактивные вкладки, чтобы освободить память для активных.

Чтобы отключить функцию, нужно:

- Открыть настройки Chrome. Можно сделать это, введя в адресной строке URL: `chrome://settings/`;
- Перейти в настройки «Производительность». Для этого на левой панели страницы настроек нужно найти список доступных опций и выбрать «Производительность»;
- Отключить «Экономия памяти». Для этого нужно нажать на переключатель.

После отключения функции вкладки не будут переходить в спящий режим.

Также в Chrome в той же настройке «Производительность» в опции «Сайты, которые всегда активны» нужно добавить адрес web-сервера.



## 12 Организация перехода со схемы в Web-версию «TrendClient Web» и обратно

### 12.1 Основные положения

Архивы ведутся только в NordWind. Чтобы иметь возможность просматривать архивы NordWind должна быть запущена программа NordWind\_Arch\_Server. Архивы SimInTech через блок Web-сервер просмотреть нельзя.

Узнайте TCP-порт, который использует NordWind\_Arch\_Server. Введите его в свойство «Порт TrendClient» (trendclient\_port) в окне «Параметры проекта» на вкладке «Параметры расчета» в секции «TrendClient».

### 12.1 Переход в архив по кнопке

Переход в Web-версию «TrendClient Web» для просмотра архива можно сделать с помощью блока «ИНТЕРФЕЙС – Кнопка». В свойстве «Действие» (where\_to\_go) кнопки выберите пункт «Перейти по ссылке». В свойстве «Ссылка, на которую перейти» (link\_go) напишите строку `http://$host$:trendclient_port$/`

При генерации файлов подстрока \$trendclient\_port\$ заменится на значение свойства «Порт TrendClient». Подстрока \$host\$ заменится на текущий IP-адрес сервера, где запущены программы из набора NordWind.

При нажатии на кнопку должна открыться Web-версия «TrendClient Web». Чтобы перейти назад в схему нажмите на кнопку «назад», которая располагается в верхнем левом углу Web-страницы «TrendClient Web».

### 12.1 Переход в архив через блоки «ИНТЕРФЕЙС - Аналоговый датчик» и «ИНТЕРФЕЙС – Дискретный»

В блоках «ИНТЕРФЕЙС - Аналоговый датчик» и «ИНТЕРФЕЙС – Дискретный» можно организовать переход в Web-версию «TrendClient Web». При этом в списке выбранных сигналов уже будут сигналы, отображаемые этим блоком.

В свойстве «Управление оператором» (control\_operator) выберите пункт «Несколько нестандартных окон управления». В свойстве «Нестандартные окна управления (имя проекта; описание)» (control\_windows) напишите строку:

```
trendclient;Посмотреть архив $self.signal_name$_$self.prefix$::trendclient_port$/index.html?name=$self.signal_name$_$self.prefix$
```



При клике по блоку появится окно со строкой «Посмотреть архив 701trt005\_value». Далее нужно кликнуть по этой строке. Должна открыться Web-версия «TrendClient Web» с 701trt005\_value в списке выбранных сигналов.

При генерации файлов подстрока \$self.signal\_name\$\_\$self.prefix\$ заменится на полное имя сигнала. \$trendclient\_port\$ заменится на значение свойства «Порт TrendClient». В конце ссылки после name= пишутся сигналы, которые будут в списке выбранных сигналов «TrendClient Web». Можно написать несколько сигналов через двоеточие, например:

trendclient;Посмотреть архив \$self.signal\_name\$\_\$self.prefix\$::\$trendclient\_port\$/index.html?name=\$self.signal\_name\$\_\$self.prefix\$\_k1:\$self.signal\_name\$\_\$self.prefix\$\_k2

Например, если имя сигнала у блока было 701trt005\_value, то после преобразований будет открываться ссылка:

[http://192.168.56.112:8080/index.html?name=701trt005\\_value\\_k1:701trt005\\_value\\_k2](http://192.168.56.112:8080/index.html?name=701trt005_value_k1:701trt005_value_k2)

