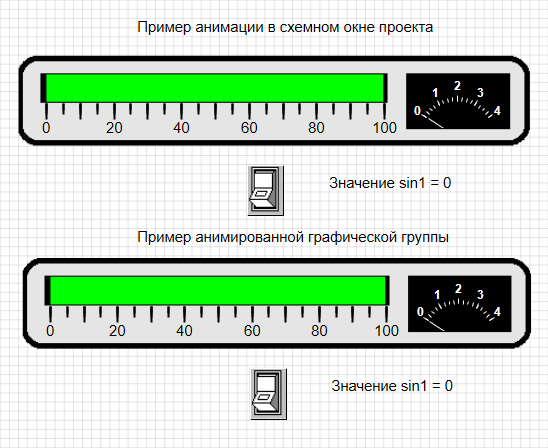
**Пример создания анимированных объектов**

В данном примере рассмотрен процесс создания внешне идентичных анимированных объектов SimInTech двумя различными способами:

* добавлением графических примитивов непосредственно в схемное окно проекта;
* созданием составного объекта, обособленного в контейнере «графическая группа».

На рисунке ниже представлен внешний вид этих объектов.



Внешне идентичные анимированные объекты: а – объект, скомпонованный из примитивов непосредственно в **схемном окне проекта**; б – объект, скомпонованный внутри контейнера «графическая группа»

В отдельных примерах рассмотрены случаи создания анимированной пиктограммы для блока «Субмодель» и панели управления.

Процесс создания анимированного объекта условно можно разделить на три этапа:

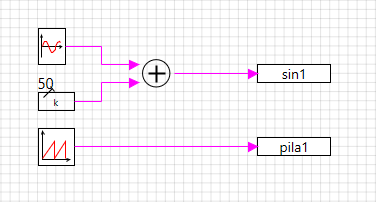
1. Компоновка изображения для будущей анимации.
2. Создание необходимого окружения из сигналов, переменных, глобальных свойств.
3. Описание связей между созданным окружением и свойствами графических примитивов, а также логики работы.

Приведённый порядок следования этапов не является обязательным, действия могут выполняться в удобной для разработчика последовательности.

Для любой анимации необходимы динамически изменяющиеся в процессе расчёта значения. В реальном проекте моделирования такие значения есть в наличии де-факто, если появляется задача их визуализации. В нашем примере пока таких сигналов нет, поэтому мы начнём с нулевого этапа – создания модели, порождающей какие-либо величины, изменяющиеся во времени.

*Нулевой этап – создание модели, предмета визуализации*

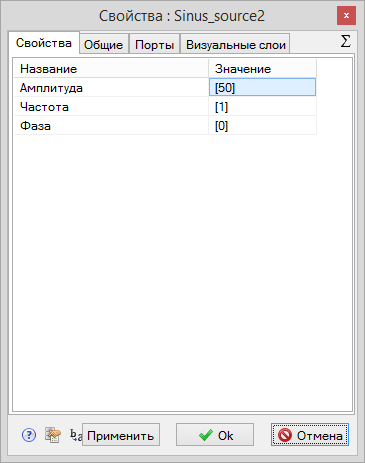
Для этого создадим новый пустой проект типа «Схема автоматики» (процесс создания анимированных объектов во всех типах проектов одинаков). Затем соберём в нём следующую модель на основе библиотечных блоков SimInTech.

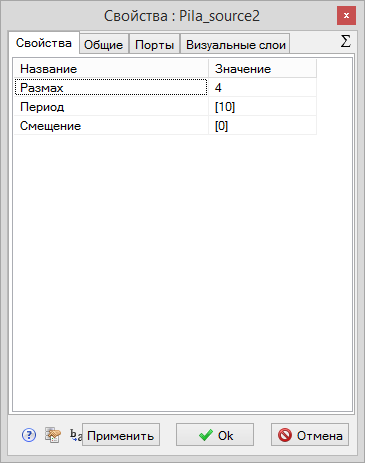


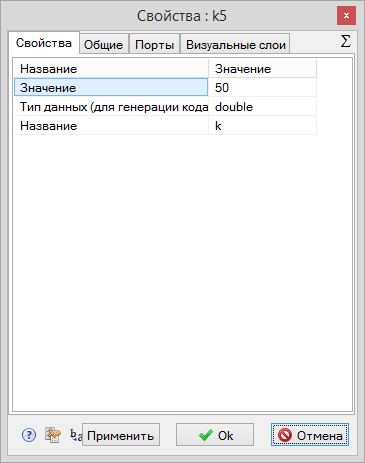
Исходная расчетная модель

Изменим свойства блоков источников следующим образом:

* синусоидальный источник: амплитуда = 50;
* константа = 50;
* пилообразный источник: размах = 4.







Характеристики динамических блоков модели

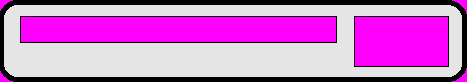
При запуске расчёта данной модели на выходе мы будем иметь две динамически изменяющихся величины: синусоидальный сигнал с диапазоном значений от 0 до 100 и пилообразный периодический сигнал с диапазоном значений от 0 до 4. Теперь можно приступать к созданию анимированного виртуального показывающего прибора, который будет осуществлять визуализацию рассчитываемых сигналов.

*Первый этап – компоновка изображения для будущей анимации*

Мы рассмотрим процесс создания виртуального показывающего прибора с двумя шкалами и одним цифровым показанием. Данный прибор будет создан в двух вариантах: как набор примитивов в схемном окне проекта и как содержимое графической группы.

**Виртуальный прибор, скомпонованный непосредственно в схемном окне проекта**

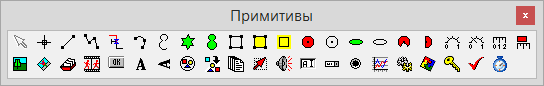
1. Сначала в любом доступном редакторе графики создадим растровое изображение лицевой панели нашего будущего прибора.



Растровое изображение лицевой панели будущего виртуального прибора

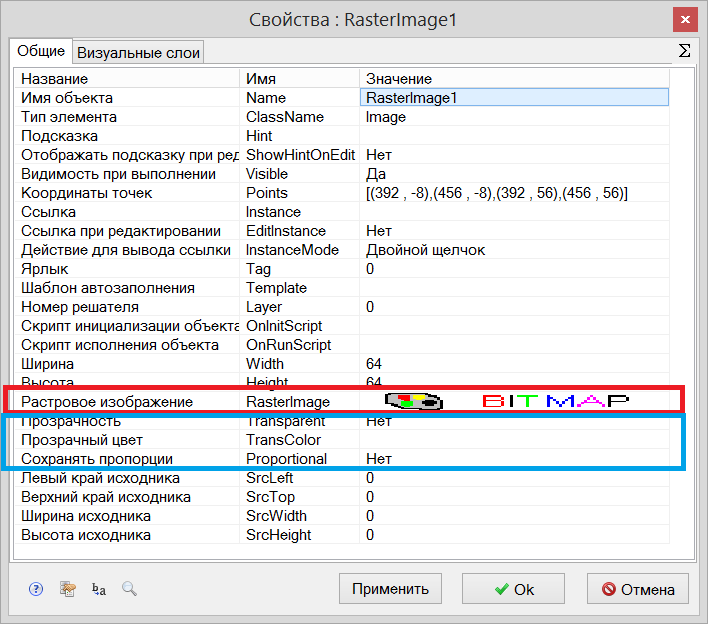
Пурпурный цвет (RGB: 255 0 255) используется в тех местах, которые в окончательном варианте должны стать прозрачными. Можно использовать любой другой цвет, однако использование пурпурного для этих целей является традиционным. Сохраняем полученный рисунок в одном из растровых форматов: BMP, JPG, PNG. Дальнейшие действия выполняются в окне созданного ранее проекта SimInTech.

1. Для продолжения нам понадобятся графические примитивы. Панель с примитивами можно вызвать из меню главного окна SimInTech: **«ГО: Вставка→ Панель примитивов...»**.



Панель графических примитивов SimInTech

1. Для добавления созданного изображения лицевой панели нашего прибора в схемное окно проекта нужно использовать графический примитив «Растровое изображение». После его добавления в схемное окно проекта нужно вызвать окно свойств примитива по двойному клику на нём и в свойстве «Растровое изображение / RasterImage» указать файл с подготовленным ранее рисунком.

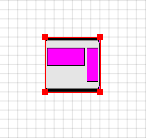


Свойства примитива «Растровое изображение / RasterImage»

1. Затем нужно установить следующие свойства примитива:

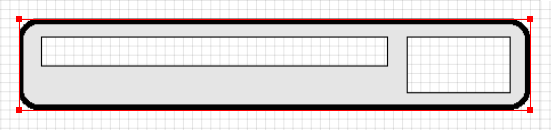
* «Прозрачность / Transparent» – Да;
* «Прозрачный цвет / TransColor» – выбрать пурпурный (RGB: 255 0 255);
* «Сохранять пропорции / Proportional» – Да.

По завершении установок нажать в окне «Свойства» кнопку «Ок». В результате в схемном окне проекта должно появиться нечто похожее на изображение ниже.



Вид примитива «Растровое изображение / RasterImage» после предварительной настройки

Изменяя размеры примитива можно добиться более приемлемого отображения.



Вид примитива «Растровое изображение / RasterImage» после изменения размеров

1. Добавим примитив «Полигон», который будет выполнять роль шкалы, заполняющейся по значениям от синусоидального источника. Полигон должен иметь вид прямоугольника. Для единообразия ваших последующих расчётов с примером рекомендуется первую точку полигона размещать в верхнем левом углу и далее ставить точки по часовой стрелке.



Взаимное расположение вершин полигона

Пусть полигон перекрывает по высоте длинную щель в лицевой панели нашего виртуального прибора.



Добавление полигона

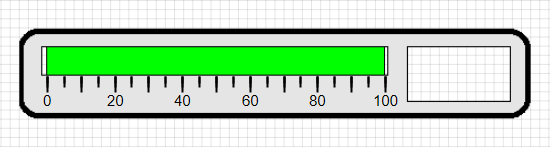
1. Затем нужно переместить получившийся полигон за растровую картинку. Это можно сделать, например, выделив полигон, и в контекстном меню, вызванном по правой кнопке мыши, выбрать пункт « Поместить назад». Полигон отобразится за растровым изображением после того, как с него будет снято выделение.

Обратите внимание: любой выделяемый графический примитив отображается поверх других независимо от его реального положения. После снятия выделения примитив отображается в соответствии со своим порядком размещения по глубине. Подробнее о манипуляциях с графическими объектами читайте в соответствующем разделе.

1. Следующим шагом нанесём градуировку на шкальную часть нашего прибора. Для этого проще всего использовать примитив « Линейная шкала». Разместим его под щелью с полигоном, выровняем шкалу так, чтобы её крайние риски совпадали с левой и правой границами полигона и установим следующие свойства:

* «Число делений на шаг / DivCount» – 4;
* «Толщина малого штриха / SmallWidth» – 2;
* «Толщина большого штриха / LargeWidth» – 2.

1. В итоге изображение должно стать похожим на рисунок ниже.

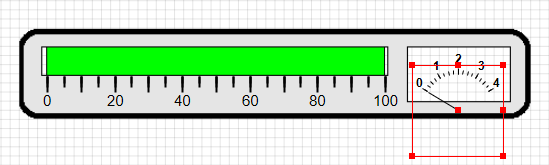


Добавление шкалы

1. Теперь добавим стрелочный прибор во второй вырез в лицевой панели виртуального прибора. Для этого будем использовать примитив « Стрелочный прибор». Установим для него следующие свойства:

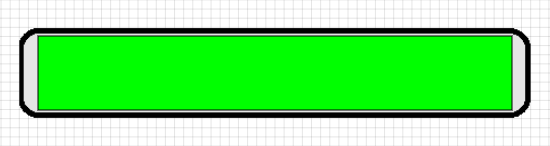
* Верхний предел – 4;
* Шаг – 1;
* Число делений на шаг – 0;
* Угол охвата шкалы (радианы) – 2;
* Число больших делений на шаг – 1;
* Шрифт – Arial, Размер – 8, Жирный.

1. Затем получившийся стрелочный прибор нужно вписать в правый вырез прибора, чтобы получилось следующее изображение. По окончании переместить примитив на задний план: контекстное меню, пункт «  Поместить назад».

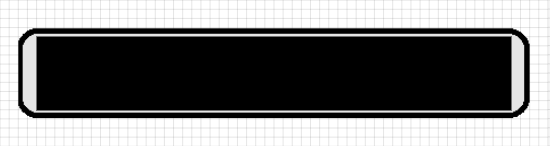


Встраивание стандартного стрелочного прибора

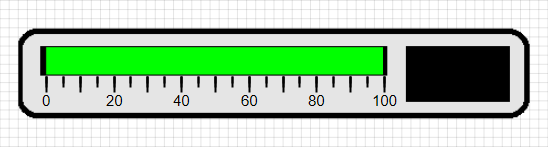
1. В качестве завершающего эстетического элемента можно разместить за прибором чёрную подложку в виде примитива « Залитый прямоугольник».



Добавление и изменение размера прямоугольника для фона



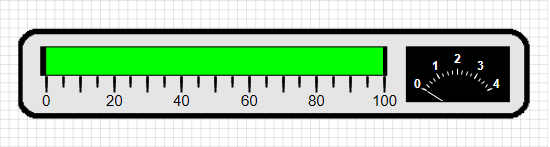
Изменение цвета заливки фонового прямоугольника на черный



Перенос фонового прямоугольника на задний план

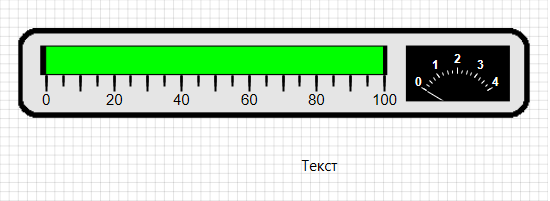
1. Для читаемости стрелочной шкалы изменим цвет её элементов на белый:

* Цвет – выбрать белый (RGB: 0 0 0);
* Цвет стрелки – выбрать белый (RGB: 0 0 0);
* Шрифт – выбрать белый (RGB: 0 0 0).



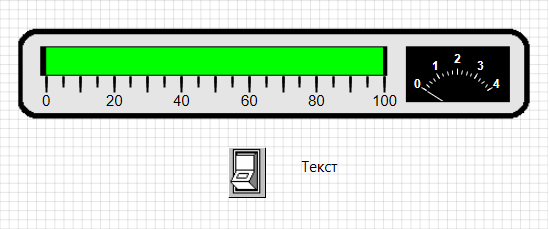
Новый внешний вид виртуального прибора

Осталось дополнить проект цифровым прибором, который будет показывать в зависимости от положения выключателя либо значение сигнала sin1, либо значение сигнала pila1. Для этого разместим в схемном окне проекта рядом с нашим виртуальным прибором примитив « Текст».



Добавление текстового примитива для отображения значений источников из модели

Оставим данный примитив без изменений и добавим управляющий графический примитив « Кнопка».

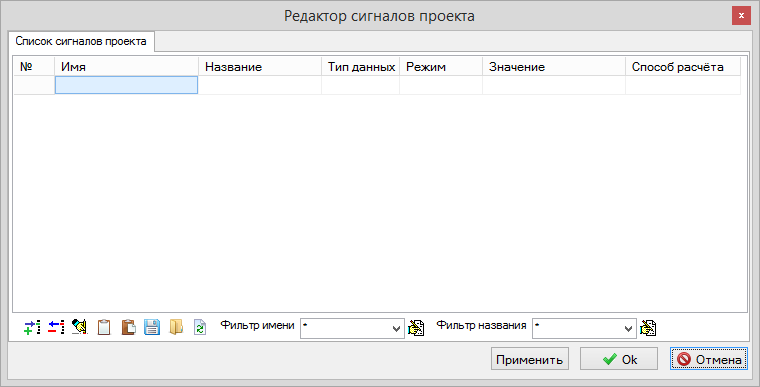


Добавление кнопки для переключения между источниками для текстового примитива

Итак, изображение в схемном окне проекта скомпоновано.

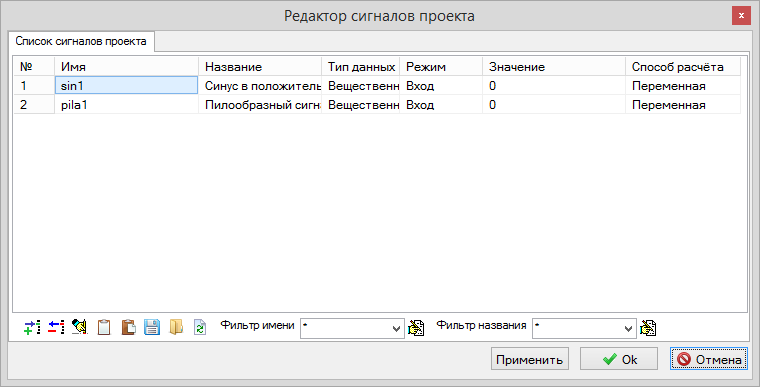
*Второй этап – создание окружения*

1. Для использования значений, рассчитываемых в модели, необходимо организовать их запись в сигналы проекта либо базы данных. Создадим два сигнала проекта. Вызовем редактор сигналов: **«ГО: Графика→Сигналы»**. Откроется окно «Редактор сигналов проекта».



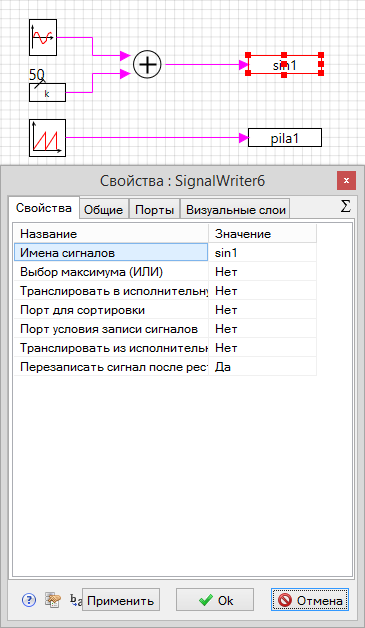
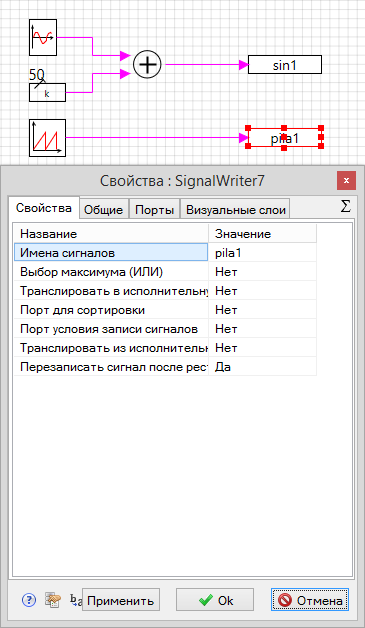
Окно редактора связей

1. В открывшемся окне по нажатию на кнопку  поочередно создадим два сигнала. По двойному щелчку в полях «Имя» и «Название» можно ввести произвольные имена и описания сигналов. Имена могут состоять из цифр и латинских букв, описания также могут содержать русские буквы. Все обращения к сигналам в SimInTech осуществляются по их именам, например, в редакторе связей или в скриптах. Установим свойства сигналов, как показано на следующем рисунке.



Настройки для добавляемых в проект сигналов

1. Теперь необходимо организовать запись расчётных величин из модели в эти сигналы. Это можно сделать с помощью библиотечного блока «Запись в список сигналов» из вкладки «Данные» главного окна. В свойстве «Имена сигналов» (вкладка «Свойства» окна «Свойства»:<Имя\_Объекта>) для каждого из этих блоков нужно вписать Имя сигнала, заготовленного для регистрации расчётной величины. Окно свойств блока вызывается по двойному клику на блоке.

Установление соответствия между выходами схемы модели и сигналами проекта

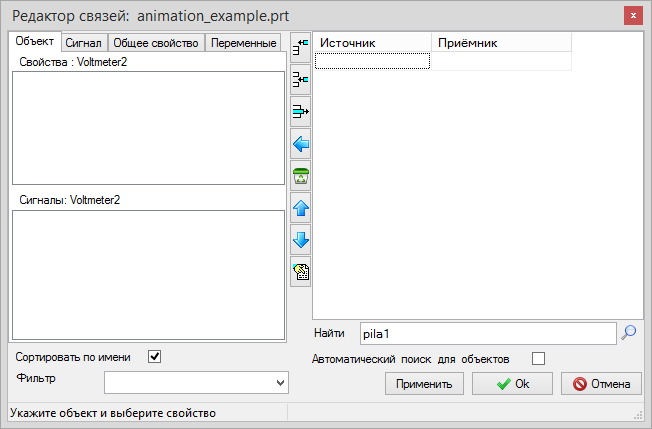
Теперь при запуске расчёта модели в сигналы sin1 и pila1 будут записываться значения синусоиды и пилы от источников из нашей схемы, собранной в схемном окне проекта.

*Третий этап – описание связей и логики работы*

Описание связей между сигналами и свойствами объектов может быть сделано как с помощью редактора связей, так и с помощью скрипта на встроенном языке программирования. В нашем примере мы будем использовать оба варианта.

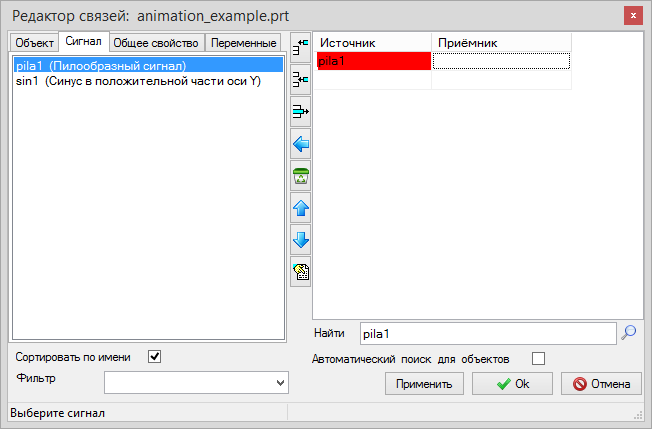
Сначала рассмотрим использование редактора связей.

1. Установим с помощью редактора связей соответствие между сигналом pila1 и показаниями стрелочной шкалы. Для этого нужно вызвать редактор связей из меню главного окна: **«ГО: Сервис→Связи...»**. Откроется окно редактора связей.



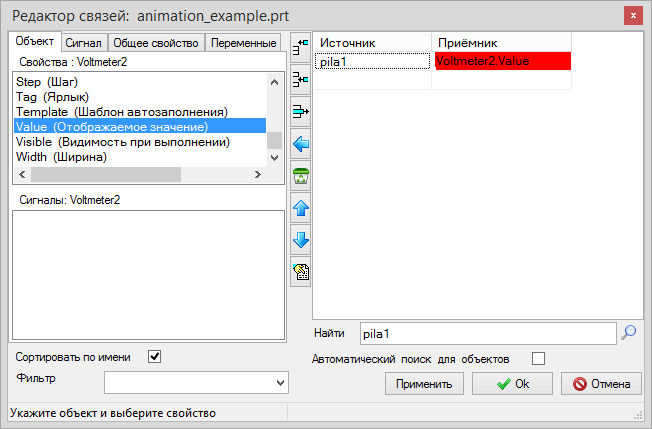
Окно редактора связей

1. Затем нужно выбрать вкладку «Сигнал» в правой части открывшегося окна: появится список сигналов проекта.
2. Сигнал pila1 нужно переместить в правую часть окна в ячейку «источник», перетянув строку с сигналом курсором мыши в эту ячейку.



Добавление сигнала pila1 в качестве источника

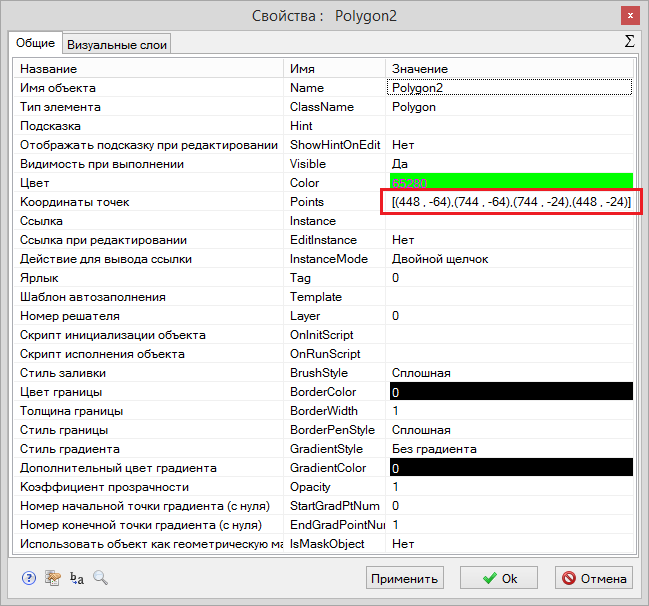
1. Затем нужно в соседней ячейке «приемник» установить свойство «Value» анимируемого примитива «Стрелочный прибор». Для этого нужно в схемном окне проекта выбрать данный примитив, а в окне редактора связей выбрать вкладку «Объект»: отобразится список свойств выбранного примитива. В открывшемся списке нужно найти свойство «Value» и перетащить его курсором мыши в ячейку «приемник» напротив заполненной ячейки «источник». Связь установлена.



Добавление свойства Value стрелочного прибора в качестве приемника сигнала pila1

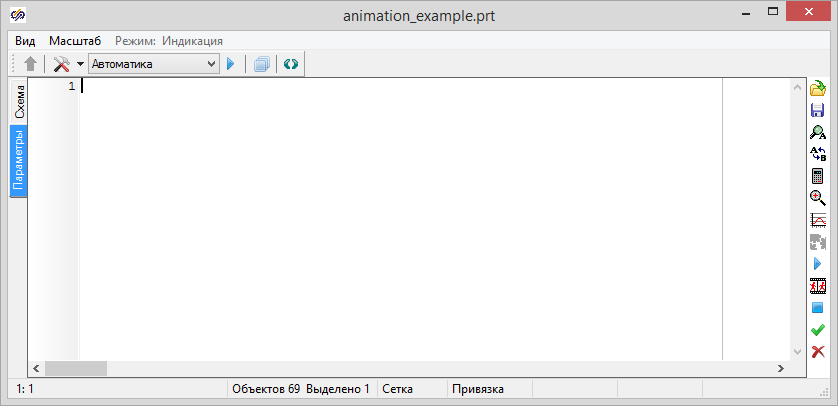
Для установления остальных связей требуются дополнительные преобразования, поэтому мы опишем их с помощью скрипта.

1. Теперь наша задача состоит в том, чтобы «привязать» ширину зелёного полигона к значению сигнала sin1 таким образом, чтобы положение правой границы полигона отмечало на шкале под ним значение, соответствующее текущему значению сигнала. Для этого нам нужно описать алгоритм изменения координат второй и третьей вершин полигона.
2. Вызовем свойства полигона двойным кликом ЛКМ по его изображению и скопируем в буфер обмена содержимое ячейки «Points» на вкладке «Свойства».



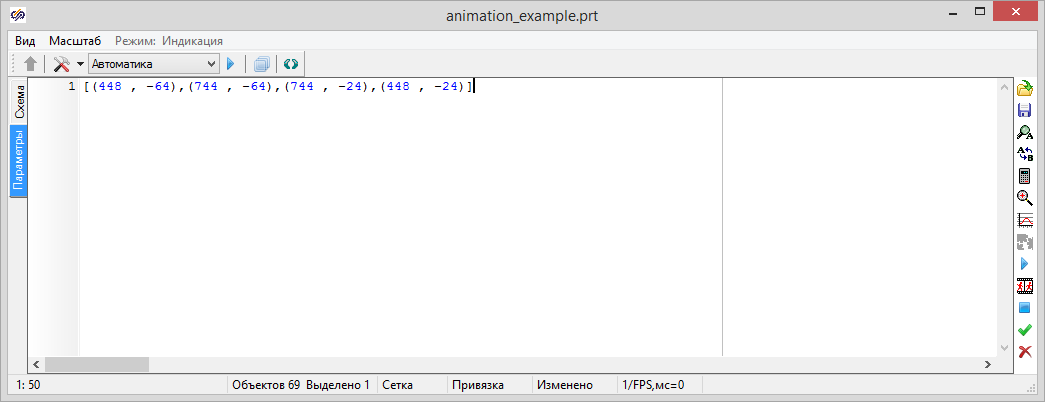
Окно свойств объекта

1. Затем закроем окно свойств, а в левой части схемного окна проекта выберем вкладку «Параметры». Откроется текстовый редактор для описания скриптов. Вставим в редактор содержимое буфера обмена. Вставится текст, содержащий координаты вершин нашего четырехугольного полигона в формате свойства «Points».

****

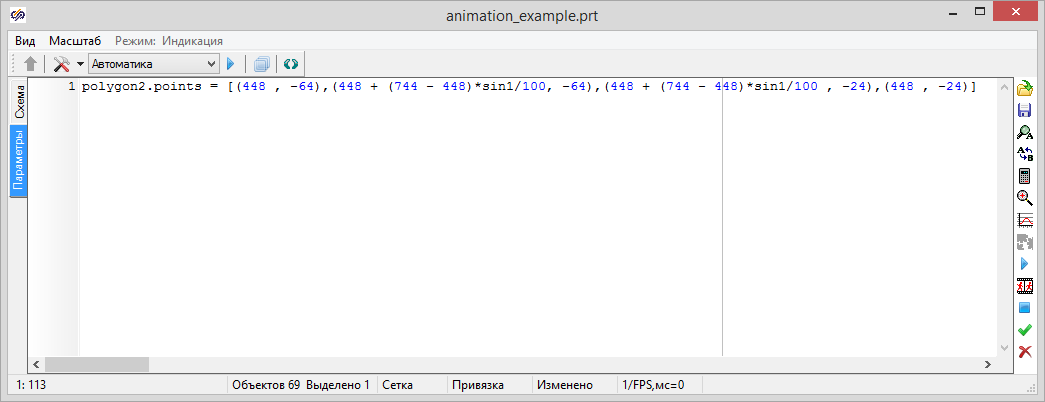
Редактор скриптов

1. По значениям координат определим координаты правой верхней и правой нижней вершин. В нашем случае это вторая и третья пара координат. Очевидно для перемещения правой вертикальной границы полигона нам нужно задавать одинаковые значения координаты Х в этих парах. Также необходимо привести диапазон значений сигнала sin1 к диапазону координат Х. Для этого нужно разделить диапазон координат Х на диапазон значений сигнала sin1. Полученное значение будет равно приращению координаты Х, соответствующему единичному значению сигнала sin1. Итак, описание координат нужно привести к следующему виду:



Формирование динамически изменяющихся координат для полигона

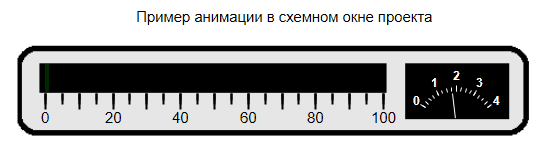
1. Теперь осталось присвоить свойству Points нашего полигона получившееся динамически рассчитываемое выражение. Для этого дополним его операцией присвоения.



Окончательный вид строки скрипта для привязки размера шкалы из полигона к значению источника sin1

Как видно из примера формат записи свойств графических объектов в скрипте формируется в манере языков объектно-ориентированного программирования: <имя\_графического\_объекта>.<название\_свойства> .

1. Чтобы проверить работу созданной нами анимации вернемся на вкладку «Схема» в схемном окне проекта и переведём окно в режим «Индикация» нажатием кнопки на панели инструментов схемного окна. Затем запустим процесс расчета модели нажатием кнопки «Пуск» в панели инструментов ГО. Анимация должна выглядеть примерно так.



**Пример анимации на текущей стадии**

1. Теперь дополним наш скрипт логикой, описывающей механизм переключения значения, отображаемого текстовым примитивом, с помощью примитива «Кнопка».

**if** button1.down **then**

**begin**

textlabel4.text = "Значение pila1 = ";

textlabel4.values = pila1;

**end**

**else**

**begin**

textlabel4.text = "Значение sin1 = ";

textlabel4.values = sin1;

**end**

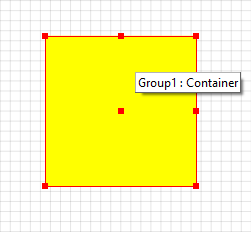
Обратите внимание: свойство «Text» примитива «Текст / Textlabel» предназначено для хранения статичных надписей. Динамически изменяющиеся значения должны присваиваться свойству «Values».

Свойство button1.down принимает значение TRUE при нажатой кнопке и FALSE при отжатой. На основании значения этого свойства свойствам текстового примитива textlabel4.text и textlabel4.values присваиваются соответственно текстовое описание и значение либо для сигнала pila1, либо для сигнала sin1.

Вновь запустим процесс расчета модели: цифровой индикатор должен отображать значения в соответствии с положением переключателя. Сам переключатель должен срабатывать по клику ЛКМ.

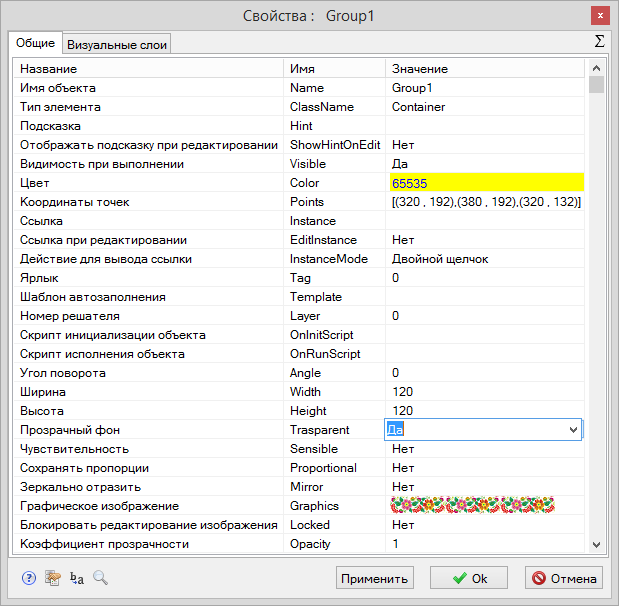
**Виртуальный прибор, скомпонованный в графическом контейнере**

1. Теперь соберём аналогичный виртуальный показывающий прибор внутри графического контейнера. Для наглядности в качестве контейнера мы используем графический примитив «Группа», который разместим в том же проекте. Для этого выберем в панели графических примитивов объект «Группа»  и поместим его в свободное место схемного окна проекта.



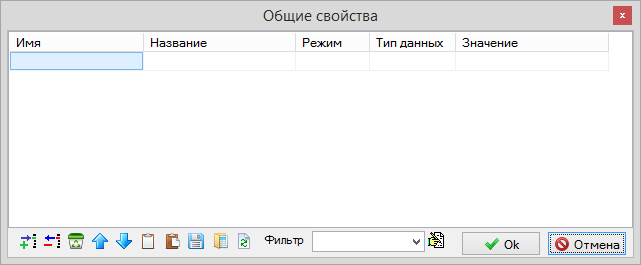
Вид вновь добавленной в схемное окно проекта пустой группы

1. Для удобства установим в свойствах группы свойство «Прозрачный фон» в значение «Да». Окно свойств группы можно вызвать из контекстного меню.



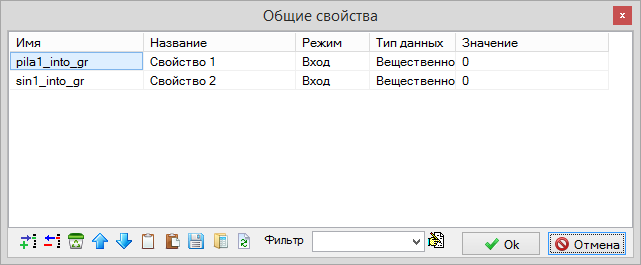
Свойства графического контейнера Group1

1. Для редактирования группы нужно дважды кликнуть на её изображении ЛКМ: откроется окно графического редактора с содержимым группы. Чтобы не отрисовывать наш показывающий прибор заново внутри группы просто скопируем его.
2. Нам осталось дополнить группу глобальными свойствами и с их помощью описать связи между сигналами проекта и свойствами графических примитивов. Для это вызовем редактор глобальных свойств из меню графического редактора: **«Сервис→Глобальные свойства...»**. Откроется показанное ниже окно



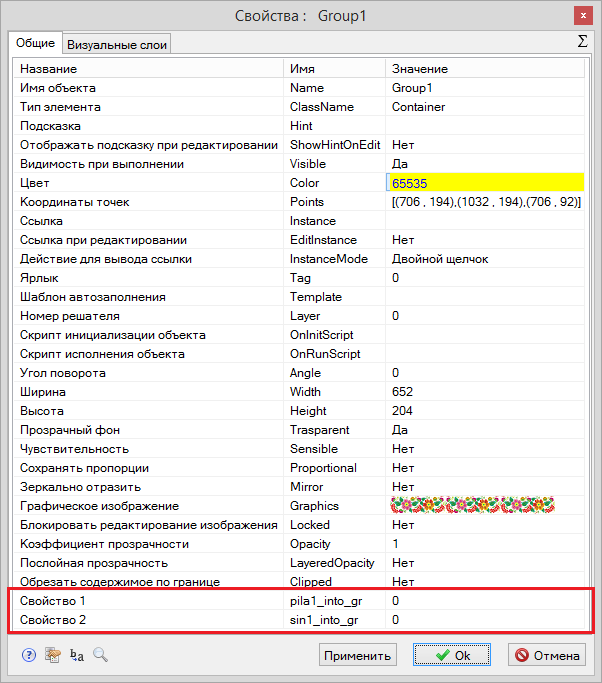
Редактор глобальных (общих) свойств

Как и в случае с предыдущим прибором нам нужно передавать в свойства примитивов значения сигналов sin1 и pila1. Поэтому подготовим два глобальных свойства для трансляции этих значений в группу.



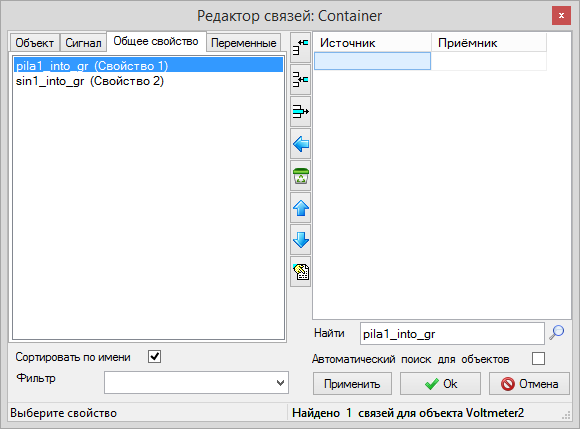
**Набор необходимых глобальных свойств и их характеристики**

1. После добавления глобальных свойств закроем это окно. Если теперь закрыть окно редактирования группы и вновь вызвать её свойства, в конце списка мы увидим два вновь добавленных свойства.



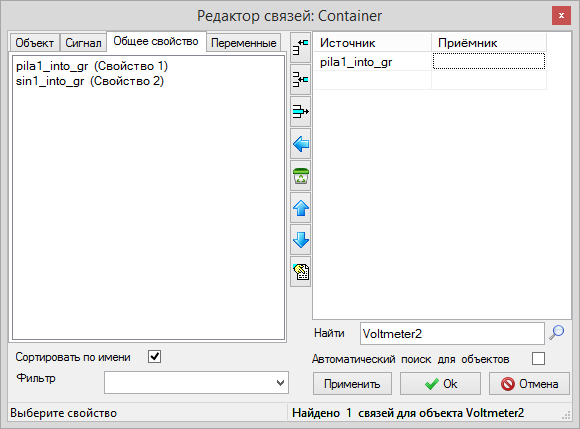
**Отображение вновь добавленных глобальных свойств графического контейнера в общем списке**

1. Последующие действия повторяют операции, описанные в п. 16 – 24 за исключением того, что окна редактора связей и редактора скриптов вызываются не из ГО, а из окна графического редактора, в котором редактируется группа. Итак, откроем редактор связей из меню графического редактора: **«Сервис→Связи...»** и выберем вкладку «Общее свойство».



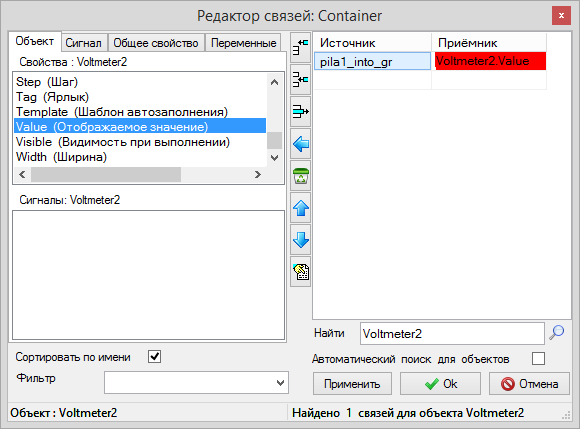
**Список глобальных свойств графического контейнера во внутреннем редакторе связей**

Перетащим свойство pila1\_into\_gr в ячейку «Источник» в правой части окна, как мы поступали до этого с сигналом проекта.



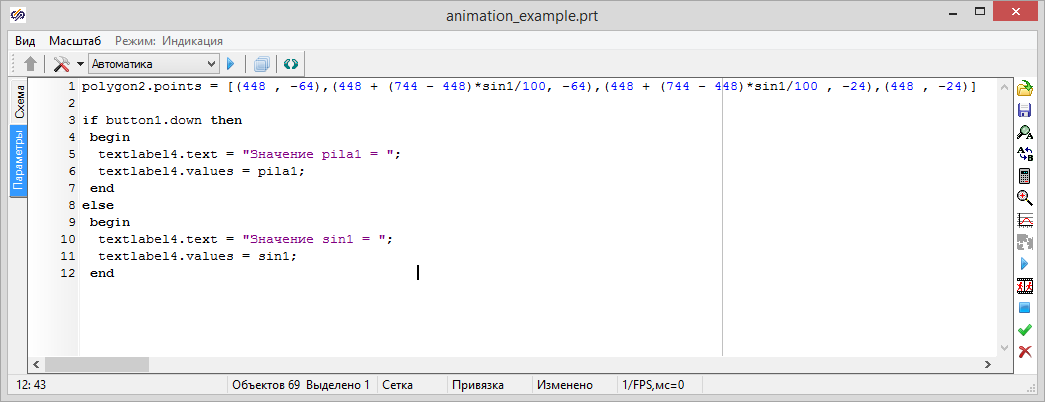
**Выбор глобального свойства pila\_into\_gr в качестве источника значений…**

Теперь выделим в окне редактирования группы примитив стрелочного прибора, чтобы во вкладке «Объект» отобразились его свойства и перетащим в ячейку «Приемник» свойство «Value».



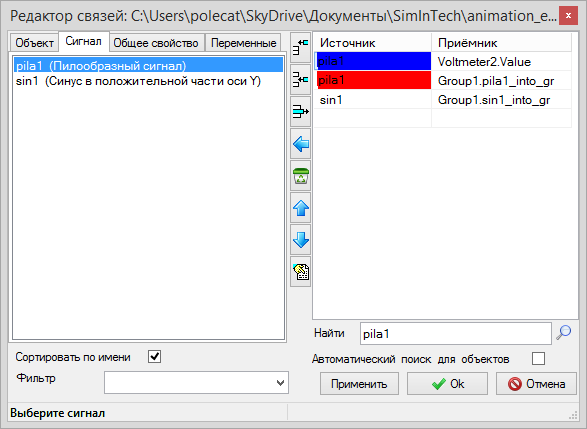
**…И свойства Value второго стрелочного прибора в качестве приемника значений**

1. В случае с индикатором в виде прямоугольного полигона повторим действия, описанные в п.23 и 24, взяв за основу координаты полигона внутри группы и используя глобальное свойство вместо сигнала. Редактор скриптов для нашего контейнера вызывается из меню графического редактора **«Сервис→Скрипт...»**. Результат должен выглядеть примерно так. Код, описывающий логику работы переключателя можно скопировать из скрипта в схемном окне проекта, поскольку наименования всех объектов внутри группы совпадают с таковыми в схемном окне, но имена сигналов нужно заменить на имена соответствующих им глобальных свойств.

****

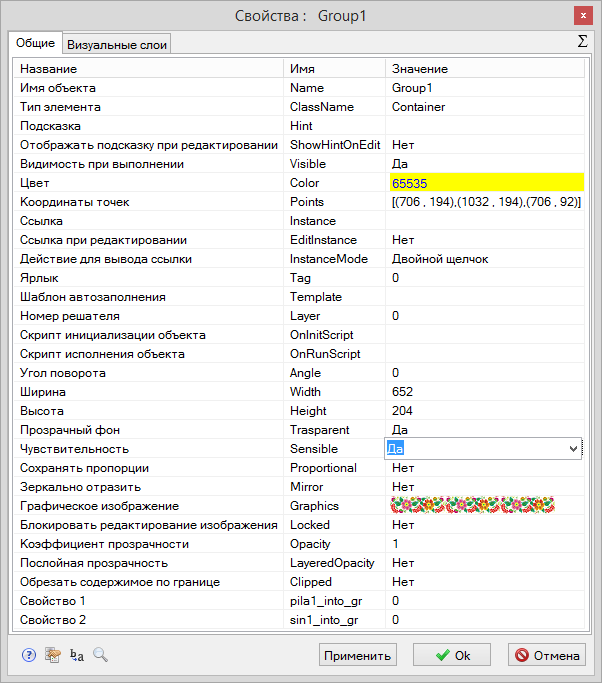
**Окончательный текст скрипта**

1. На данном этапе прописаны все связи внутри группы, но ещё необходимо установить соответствие между глобальными свойствами и сигналами проекта. Чтобы сделать это, закроем окно редактирования группы и вызовем редактор связей, на этот раз из меню главного окна SimInTech: **«ГО: Сервис→Связи...»**. Для отображения нужных свойств выделим в схемном окне проекта нашу группу. Установим следующее соответствие, последовательно перетаскивая в правую часть окна свойства и сигналы.



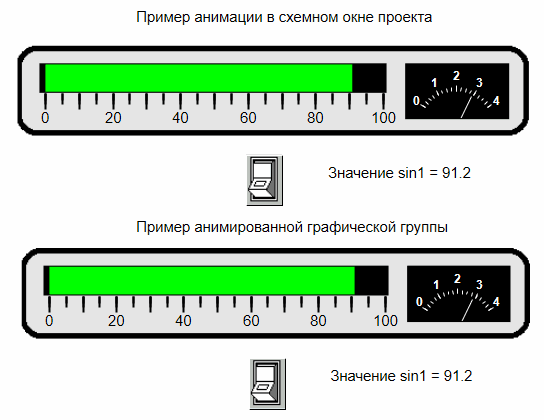
**Полный набор связей внутри графического контейнера**

1. Для того, чтобы переключатель внутри группы реагировал на нажатия мышки нужно установить свойство группы «Чувствительность» в значение «Да».



**Свойство «Чувствительность / Sensible» в значении «Да»**

1. Теперь установление соответствия между сигналами проекта и содержимым группы завершено. Можно запустить расчет и убедиться в идентичности показаний двух наших приборов.



**Вид готовых анимированных объектов**