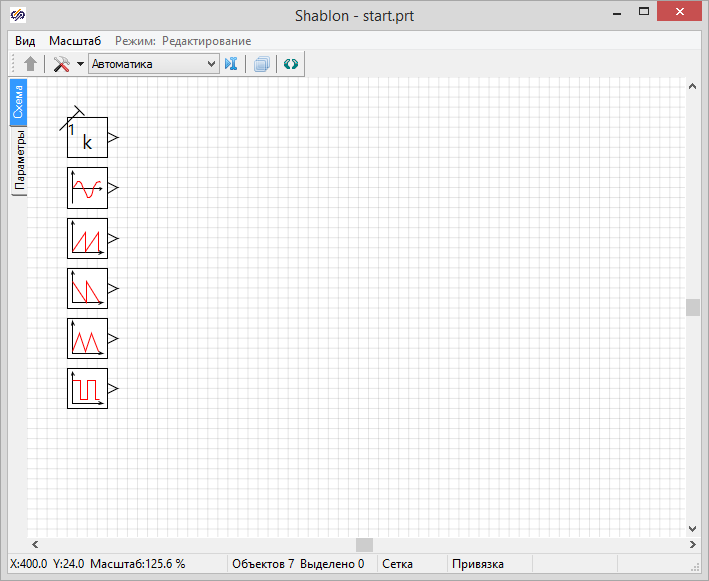
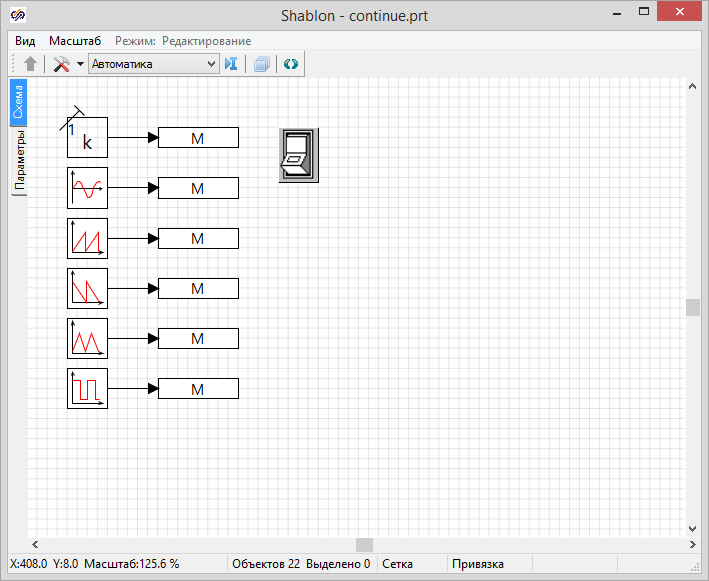
**Пример использования шаблона**

В данном примере показан вариант использования шаблона в проекте, содержащем несколько источников сигналов. Исходный состав проекта включает в себя шесть источников. Наша задача будет состоять в том, чтобы с помощью текстовых примитивов отображать текущие значения, получаемые от этих источников. При этом для каждого источника должен использоваться свой текстовый примитив, а отображение должно включаться и отключаться по кнопке.



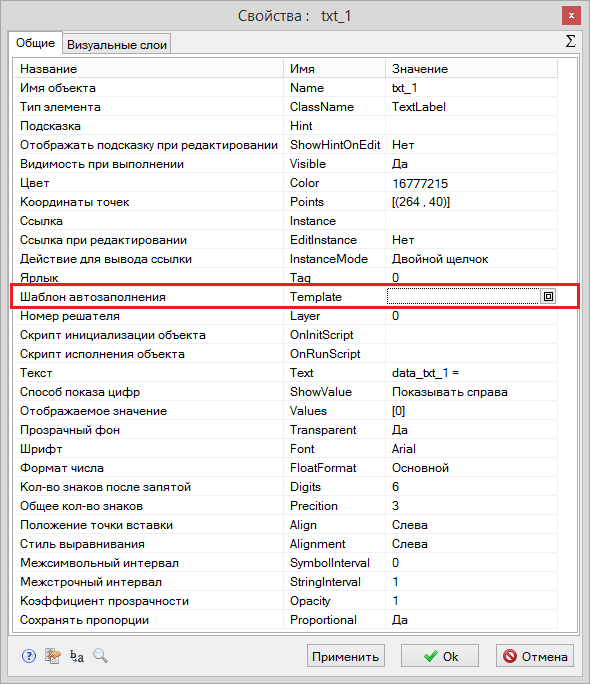
**Исходный вид проекта**

Сначала добавим необходимые блоки в проект. В первую очередь добавим приемники значений источников и соединим их с блоками источников. Для этого в палитре компонентов Схемного окна откроем вкладку «Данные», а в ней выберем блок «Запись в список сигналов» и добавим его в Схемное окно. Затем соединим выход первого источника и вход данного блока. Повторим данную операцию для оставшихся источников. Также добавим примитив «Кнопка» с панели примитивов. Промежуточный результат должен выглядеть аналогично иллюстрации ниже.

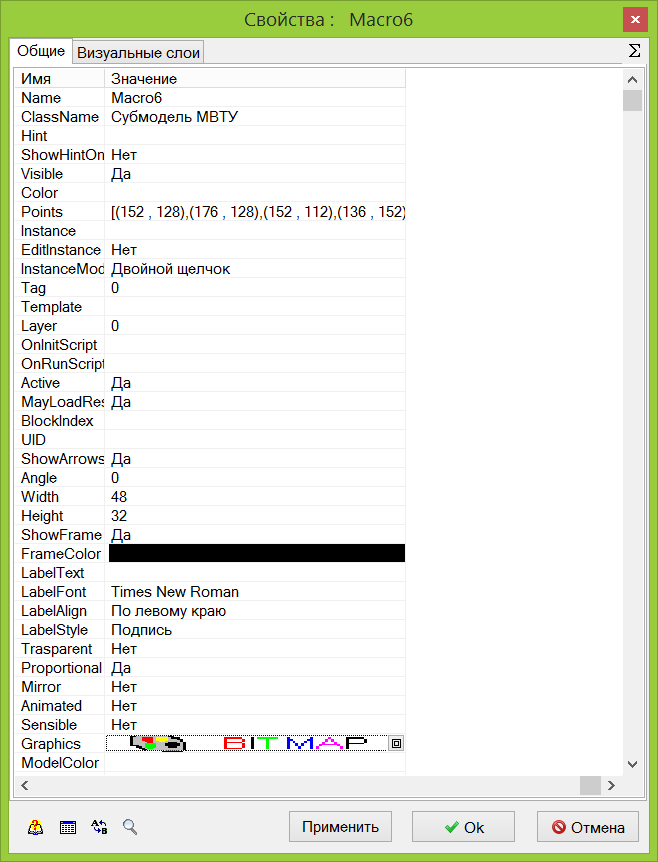


**Организация записи генерируемых значений в сигналы проекта**

Добавим примитив «Текст» из панели примитивов, вызвав её через меню Схемного окна:**«СО: Вставка → Панель примитивов...»**. Затем откроем редактор свойства «Шаблон автозаполнения». Для этого по ПКМ вызовем контекстное меню и выберем пункт «Свойства объекта».



**Окно свойств исходного текстового примитива**

В открывшемся окне, во вкладке «Общие» выберем строку «Шаблон автозаполнения» и в правой ячейке нажмём кнопку . Откроется текстовый редактор, в котором мы запишем команды шаблона. С помощью шаблона мы создадим сигнал для нашего текстового примитива, опишем поясняющий текст и логику отображения значения сигнала с учётом положения кнопки с именем «Button»:

SIGNAL sig\_$self.Name$

SCRIPT $self.Name$.Text = "data\_$self.Name$ ="

SCRIPT if button.down then $self.Name$.Values = sig\_$self.Name$ else $self.Name$.Values = 0

Закроем окно текстового редактора по кнопке . Затем в окне свойств обязательно нужно нажать кнопку «Применить», иначе все изменения будут утеряны.

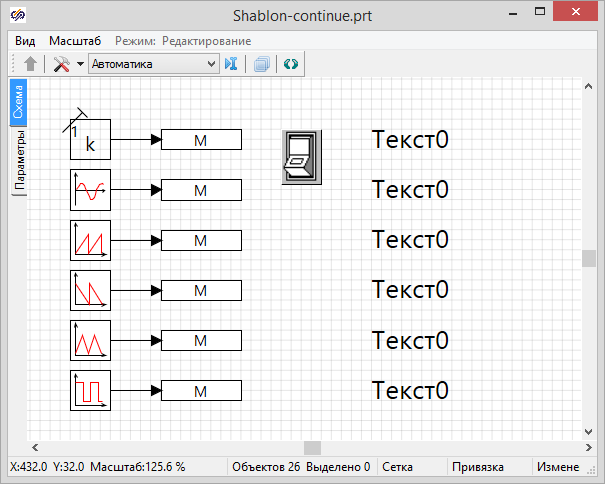
Рассмотрим подробнее команды нашего шаблона. Первая строка создаст сигнал с именем, влючающим в себя имя объекта, содержащего шаблон, т.е. в нашем случае будет создан сигнал sig\_txt\_1. Вторая строка создаст строку скрипта в Схемном окне, которая будет присваивать свойству «Text» нашего примитива строку «data\_txt\_1 =». В третьей строке задаётся условие, по которому при нажатой кнопке button свойству Values нашего текстового примитива присваиваются значения сигнала sig\_txt\_1, а при отжатой – ноль. Мы воспользовались скриптом для связывания сигнала и свойства примитива, т.к. эта связь должна подчиняться дополнительным условиям. В более простом случае, если значения сигнала всегда должны отображаться нашим текстовым примитвом, третью строку можно было бы заменить на команду CONNECT, создающую прямую связь между сигналом и свойством объекта:

CONNECT sig\_$self.Name$, $self.Name$.Values

Далее изменим имя текстового примитива с «TextLabel» на «txt\_1». Тогда при копировании данного примитива новые примитивы будут автоматически именоваться как <текст><n+1>, т.е. в нашем случае: «txt\_2», «txt\_3» и т.д.

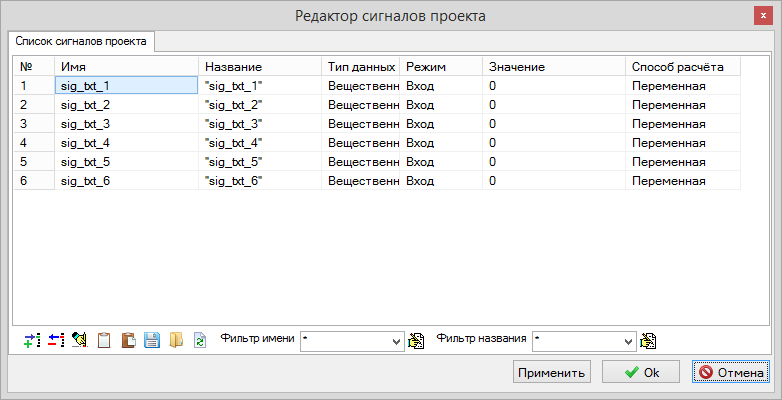
Для отображения числовых значений сигналов установим свойство «Values» в значение «Показывать справа». Теперь окно свойств нашего текстового примитива можно закрыть.

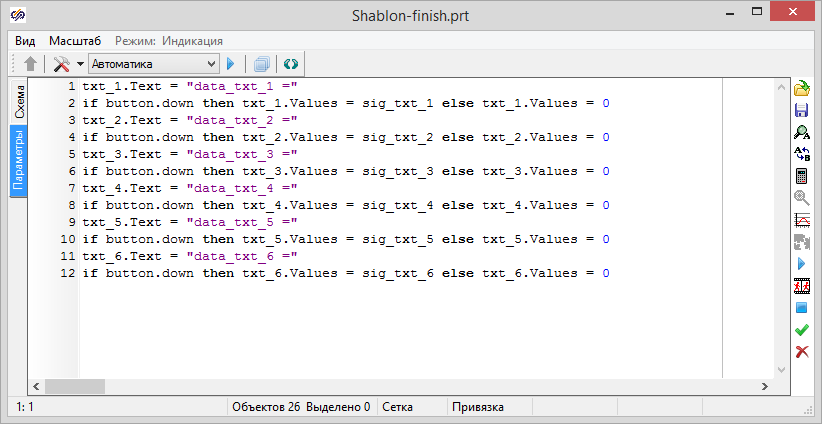
Теперь нужно размножить наш текстовый примитив с шаблоном простым копированием. Всего в проекте должно получиться шесть текстовых примитивов с именами «txt\_1», «txt\_2», «txt\_3», «txt\_4», «txt\_5», «txt\_6». На рисунке ниже представлен примерный вид Схемного окна на данном этапе.



**Вид Схемного окна после добавления текстовых примитивов**

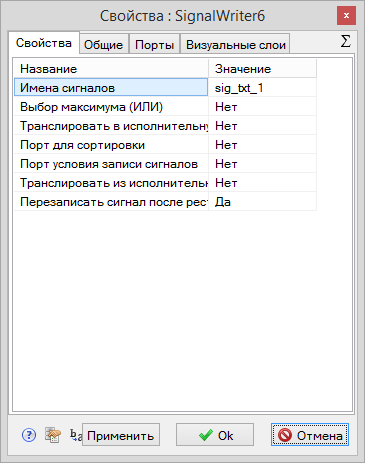
Теперь нужно воспользоваться нашим шаблоном. Для этого выделим все текстовые примитивы и в меню Схемного окна выберем пункт **«СО: Сервис → Генерация сигналов и связей»**. После этого нужно проверить список сигналов проекта и содержимое скрипта в Схемном окне. Для просмотра списка сигналов нужно выбрать пункт меню Схемного окна **«СО: Сервис → Сигналы...»**, а для проверки скрипта в левой части схемного окна выбрать вкладку «Параметры». Данные окна должны выглядеть следующим образом.





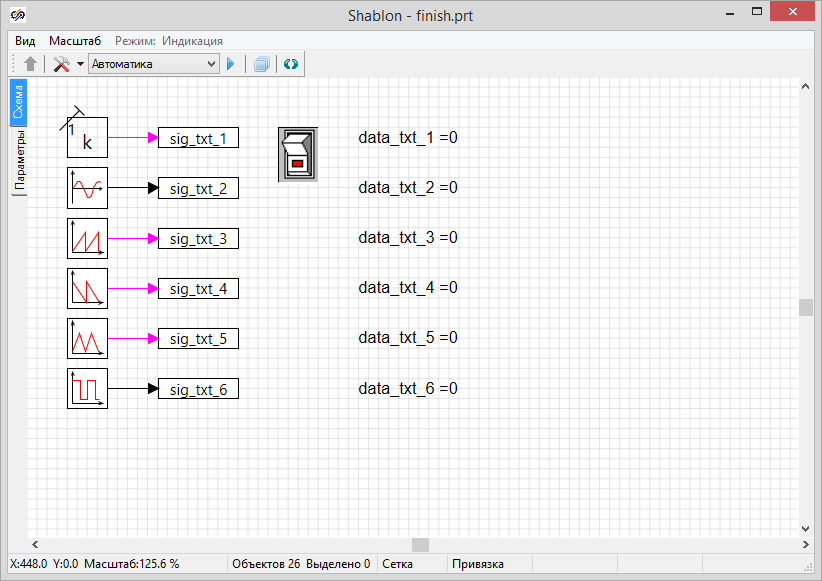
**Состав сигналов проекта и содержимое скрипта в Схемном окне после применения шаблонов**

В качестве завершающего действия необходимо указать имена сигналов в блоках «Приёмник», подключённых к источникам. Для этого нужно выделить блок, по клику ПКМ вызвать контекстное меню и выбрать пункт «Свойства объекта». В открывшемся окне выбрать вкладку «Свойства», строку «Имена сигналов» и вписать имя сигнала в ячейку справа.



**Указание имени сигнала для записи значений в блоке «Приёмник»**

Результат в Схемном окне должен выглядеть так.



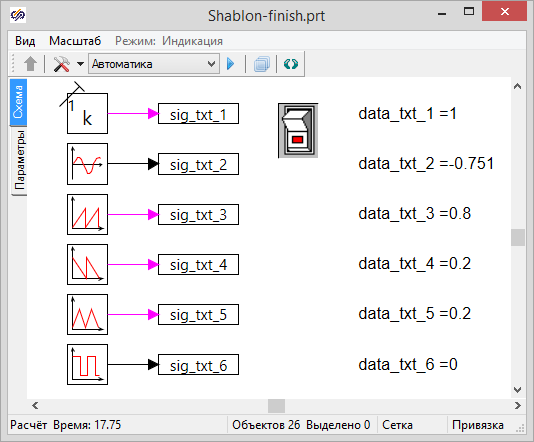
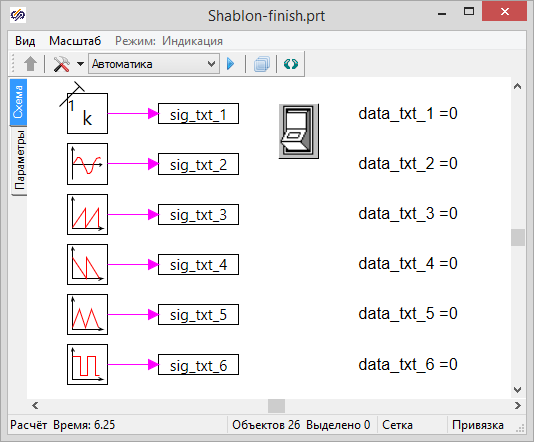
**Содержимое Схемного окна после указания сигналов в блоках «Приёмник»**

Теперь в нашем проекте можно запустить расчёт, предварительно переведя Схемное окно из режима «Редактирование» в режим «Индикация».



**Переключение режимов отображения Схемного окна**

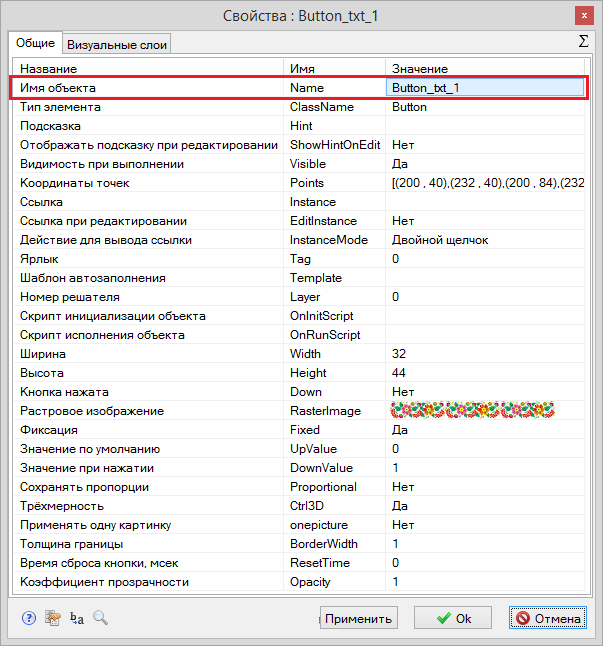
Ниже представлены виды Схемного окна при отжатой и при нажатой кнопке.



**Демонстрация работы анимации в схемном окне проекта**

Данный пример несложно модифицировать так, чтобы каждый текстовый примитив включался и отключался по своей кнопке. Сначала вернём проект на несколько шагов назад. Для этого удалим все текстовые примитивы кроме первого с именем «txt\_1», очистим список сигналов проекта и удалим все сгенерированные скрипты.

Далее переименуем кнопку из «Button» в «Button\_txt\_1» и скопируем её пять раз.



**Изменение имени примитива «Кнопка»**

В итоге проект должен содержать кнопки с именами «Button\_txt\_1», «Button\_txt\_2», «Button\_txt\_3», «Button\_txt\_4», «Button\_txt\_5», «Button\_txt\_6».

После этого отредактируем шаблон, содержащийся в примитиве «txt\_1». Заменим третью строку:

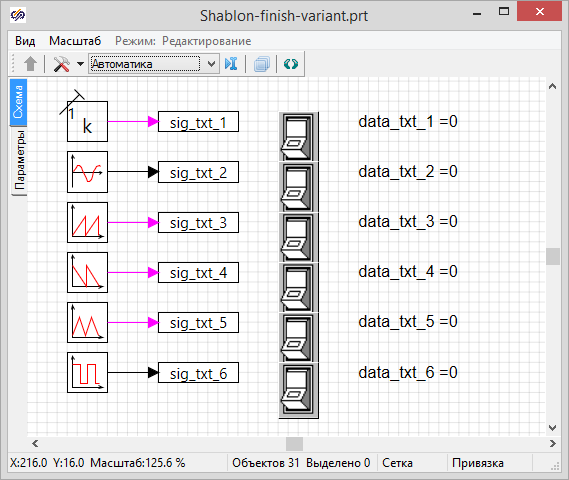
SCRIPT if button.down then $self.Name$.Values = sig\_$self.Name$ else $self.Name$.Values = 0

на строку:

SCRIPT if button**\_$self.Name$**.down then $self.Name$.Values = sig\_$self.Name$ else $self.Name$.Values = 0

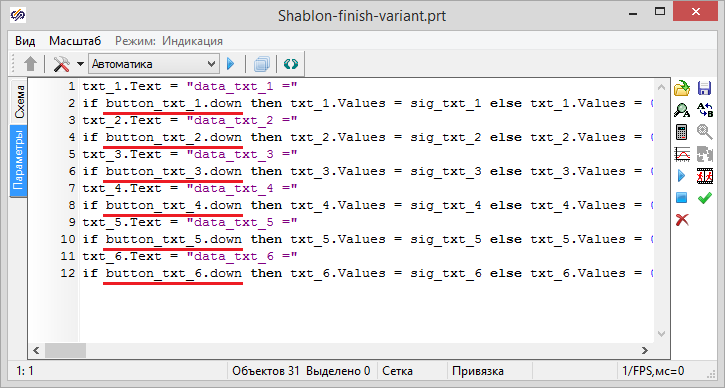
Схемное окно должно выглядетть следующим образом. На данном этапе текст в текстовых примитивах может отличаться от приведённого на изображении, это не существенно.

После этого снова размножим этот текстовый примитив, чтобы получить шесть однотипных объектов, выделим их и проведём генерацию сигналов и скриптов.



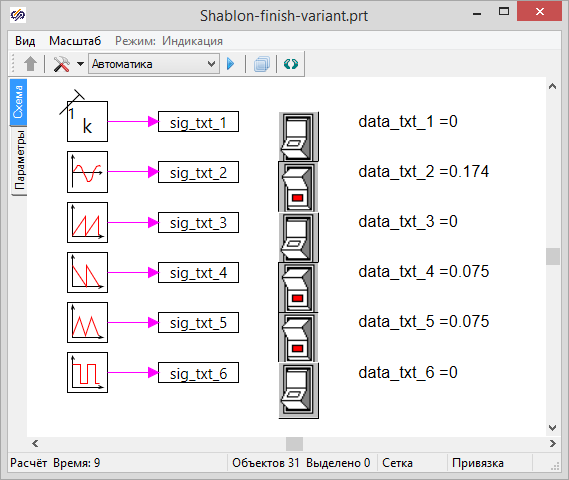
**Вид Схемного окна после модификации проекта**

Состав сигналов будет аналогичен тому, что был получен в первый раз, а текст скрипта изменится в том месте, где проверяется состояние кнопки, поскольку теперь каждому примитиву важно состояние именно его кнопки.



**Скрипт, сгенерированный по изменённому шаблону**

В итоге при запуске расчёта мы должны получить подобную картину.



**Вид Схемного окна с запущенным расчётом при использовании индивидуальных кнопок.**

Каждый текстовый примитив управляется своей собственной кнопкой.

При использовани различных языковых конструкций типа «formattext-end», «getstrformat» и т.п. можно добиться большей автоматизации при создании проекта, однако это требует более тесного знакомства с языком программирования SimInTech.