**Создание видеокадра, подключающегося к модели и расчету.**

*Возможности создания визуализационных кадров в среде SimInTech.*

Для визуализации и подачи информации пользователю в наиболее удобном для восприятия формате создаются динамические графические изображения, которые называются видеокадрами. Среда разработки SimInTech позволяет не только разрабатывать расчетные схемы, но и создавать видеокадры дли расчетных схем. Видеокадры позволяют визуализировать расчетные параметры схем в удобном для отображения виде, вывести инструменты управления (например, ключи запуска оборудования, кнопки задатчиков регуляторов) для изменения значения переменных в расчете в том объеме, который необходим конечному пользователю продукта. Видеокадры могут полностью повторить структуру пульта или панели реального объекта, что используется при создании аналитических тренажеров для различных объектов энергетического комплекса, таких как АЭС.

Видеокадры могут существовать:

1. Отдельно от модели – видеокадр в этом случае представляет собой отдельный блок, либо отдельную схему в отдельном файле;
2. В пакетном режиме - как вместе с моделью, так и отдельно от нее. В случае совместного запуска разработанные видеокадры в виде блоков либо схем находятся вместе с файлами модели в одном пакете и инициализируются при запуске модели, соответственно и обновление параметров происходит так же при запуске модели. Модель и видеокадры при этом представляют собой единый целый проект.

В случае отдельного запуска каждый видеокадр является отдельным проектом, который пользователь сохраняет в отдельном файле. При этом видеокадр подключается к модели только в тот момент, когда это нужно пользователю.

ПК SimInTech содержит инструменты разработки видеокадров, поэтому пользователь может создавать как расчетные схемы, так и кадры визуализации для этих схем. Разработка видеокадров осуществляется с помощью панели примитивов (см. **Рисунок 1**) и специально разработанных блоков, которые могут быть уникальными для каждого проекта.

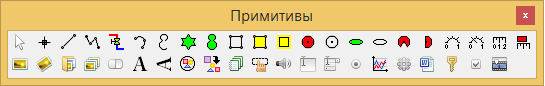


Рисунок 1. Внешний вид панели примитивов

С помощью примитивов вычерчивается «скелет» схемы, например, схема трубопроводов или схема электрической коммутации, а специализированные блоки отвечают за вывод информации (т.е. могут представлять собой визуальные имитаторы датчиков), за управление и анимацию оборудования (например, задвижек/клапанов/насосов/вентиляторов и пр.) и т.д.

Этапы разработки видеокадра:

1. Определяет содержание видеокадра (на этом этапе содержание предварительное, уточняться оно будет в процессе работы над видеокадром);
2. По предварительному содержанию создается «скелет» видеокадра при помощи примитивов и библиотечных блоков, определяется потребность в уникальных для данного видеокадра блоков, при необходимости такие блоки создаются;
3. Уникальные блоки расставляются на «скелет» видеокадра;
4. Пишутся скрипты для видеокадра (например, скрипты, для связи видеокадра с моделью, скрипты для изменения визуального отображения примитивов и пр.);
5. Уникальные блоки на кадре параметризируются. По выполнении этого этапа уже можно подключать видеокадр к модели и пошагово отлаживать его;
6. Производится окончательная проверка изображения на видеокадре и его правка при необходимости, убирается первоначальный «скелет» Видеокадр, который получится в результате завершения этого этапа, визуально будет представлять собой то, что увидит конечный пользователь продукта;
7. Окончательное тестирование блоков, проверка правильности показаний, проверка правильности анимации блоков.

Отладка (или тестирование) представляет собой проверку всех режимов работы с видеокадром и содержащимися на нем блоками, которые будут доступны конечному пользователю продукта, с целью определения правильности работы этих режимов/блоков. При этом тестируются все заложенные разработчиком функциональные возможности режимов работы и блоков.

Рассмотрим в качестве примера создание видеокадра «Многофункциональный регистратор». Данный видеокадр предназначен для визуального отображения приходящих на него значений, уставок, а также других параметров.

Итак, пройдем по этапам разработки:

1. Для начала определимся с типом визуализации отображения. Отображение информации на данном видеокадре должно быть, как цифровым, т.е. пользователь должен видеть приходящее значение в виде численной величины, так и графическим, т.е. величина должна отображаться в виде графика в зависимости от времени. Кроме того, должна быть визуализация перехода значения за уставки;
2. Создаем «скелет» блока. Первоначально «скелет» видеокадра создается при помощи примитива «Линия» / «Полилиния». После первоначального вычерчивания лист проекта выглядит таким образом (см. **Рисунок 2**):

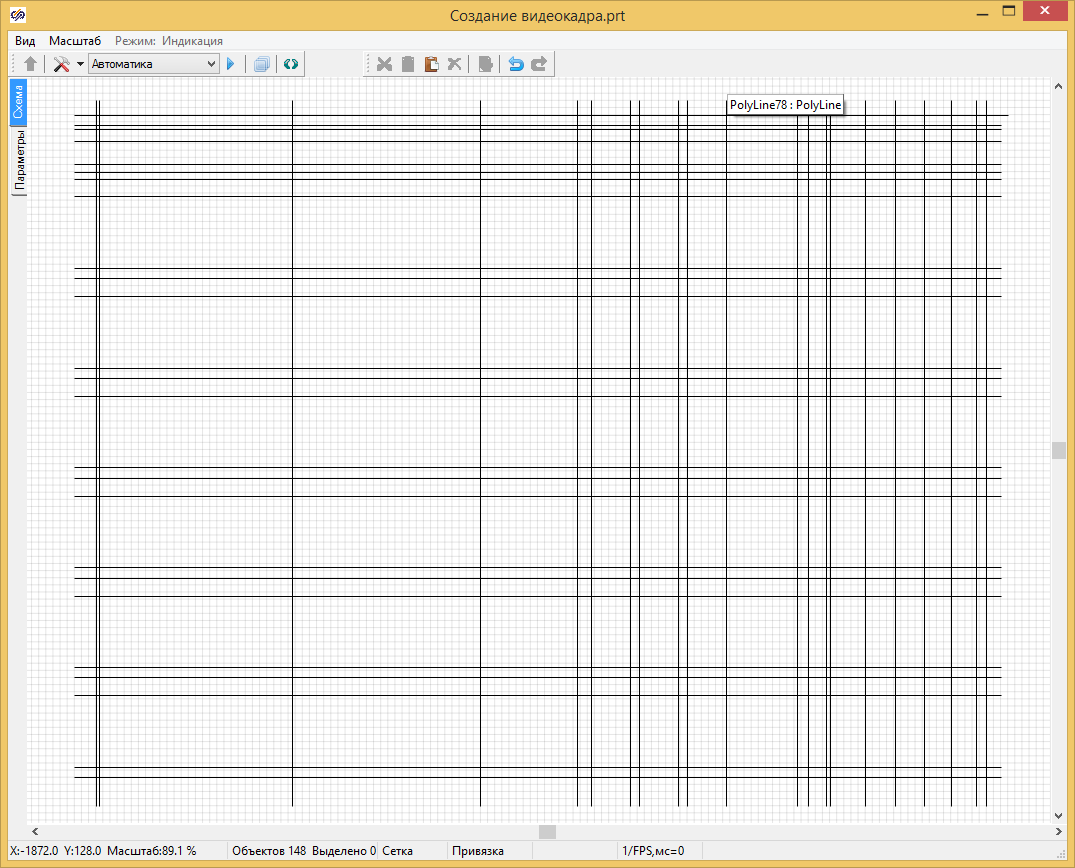


Рисунок 2. Первоначальный «скелет» блока.

Далее на данный видеокадр навешиваются примитивы, создающие «подложку» анимированных составляющих видеокадра. После наполнения примитивами скелета лист проекта выглядит так (см. **Рисунок 3**):

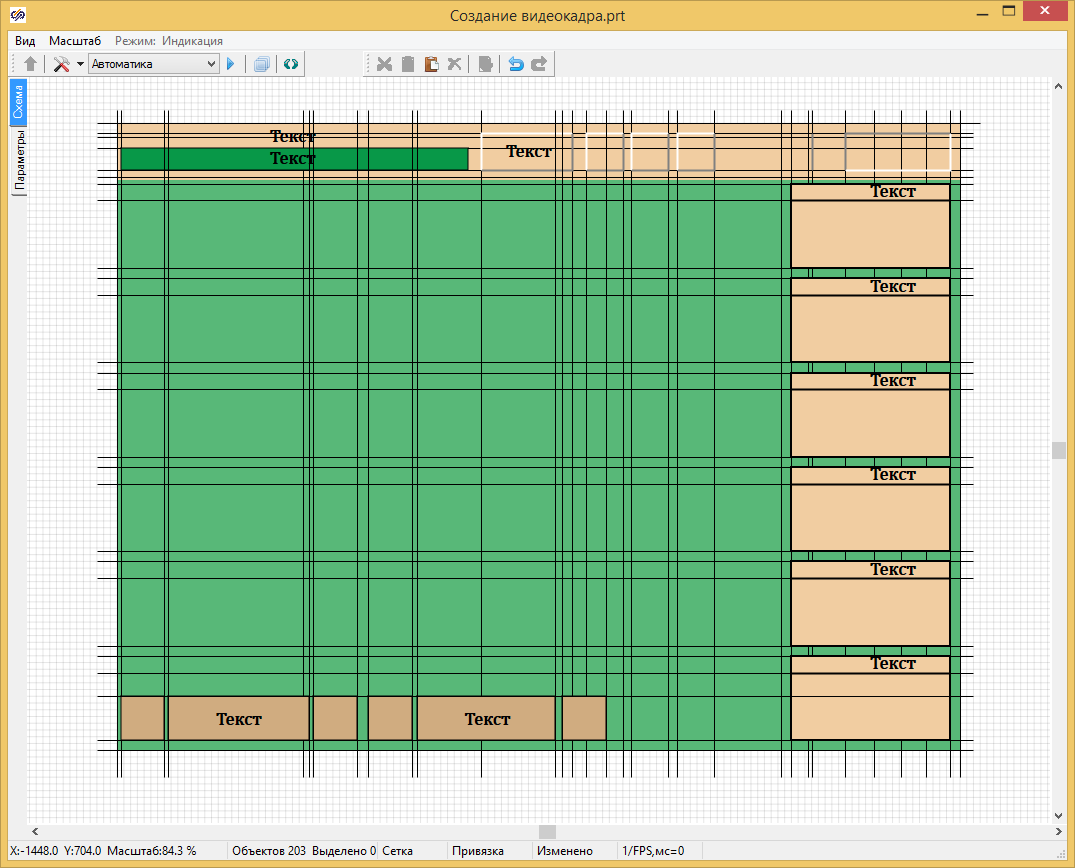


Рисунок 3. Внешний вид видеокадра после наполнения его примитивами.

На данном этапе стало ясно, что нужны два уникальных блока: первый должен отвечать за численное отображение величин, приходящих в регистратор, второй должен отвечать за построение графической зависимости величины от времени. Создаем необходимые нам блоки с внутренними скриптами;

1. Расставляем уникальные блоки на видеокадре (см. **Рисунок 4**);

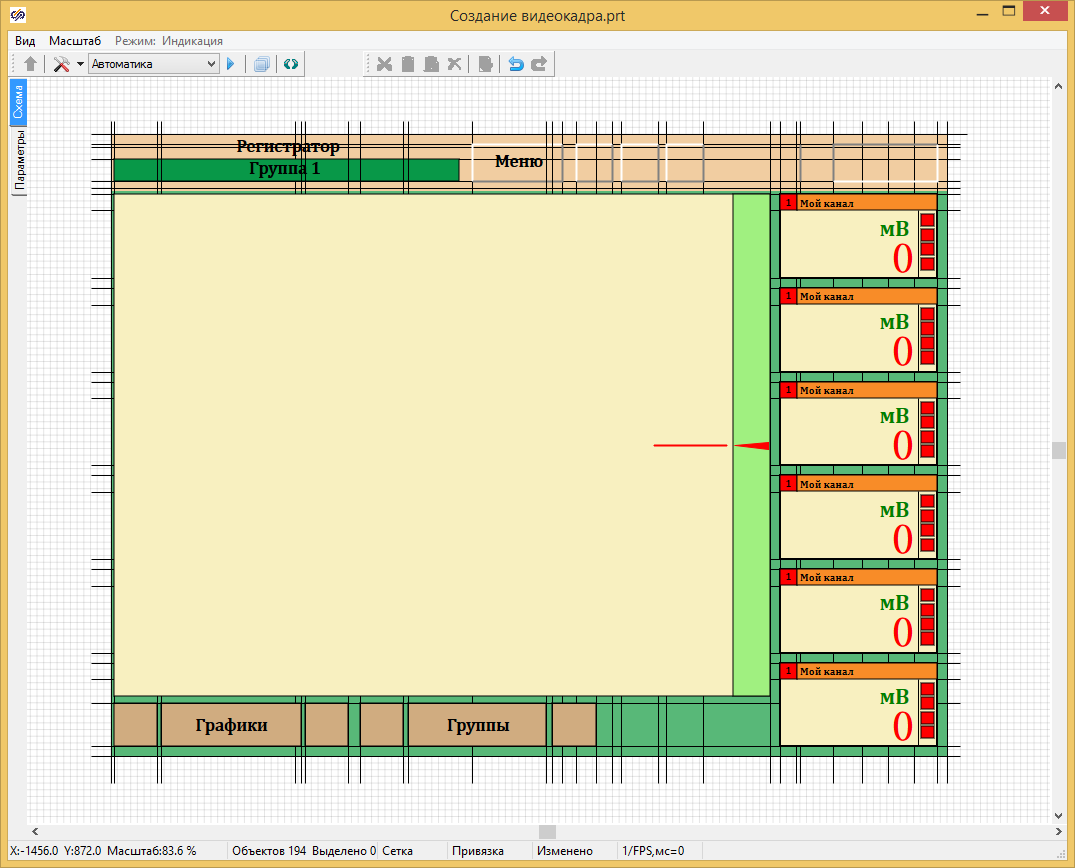


Рисунок 4. Внешний вид видеокадра после добавления уникальных блоков

1. Производим написание необходимых скриптов (см. **Рисунок 5**)

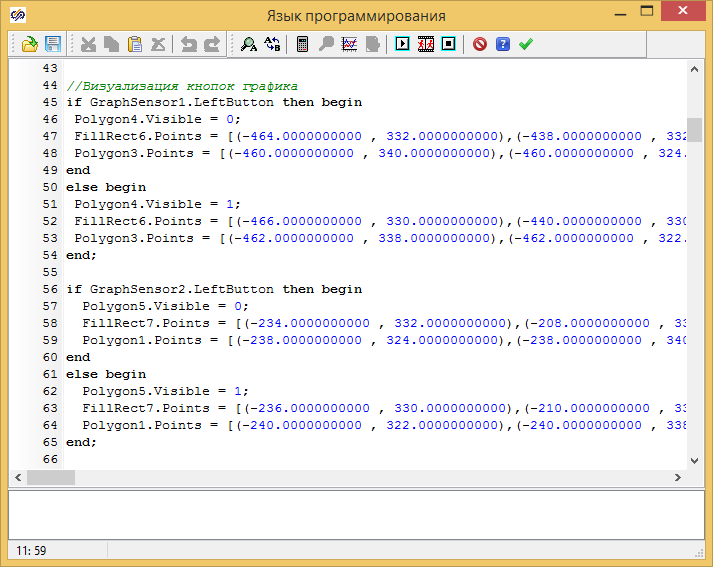


Рисунок 5. Часть скрипта, необходимого для правильной работы видеокадра

1. Параметризуем уникальные блоки (см. **Рисунок 6**).

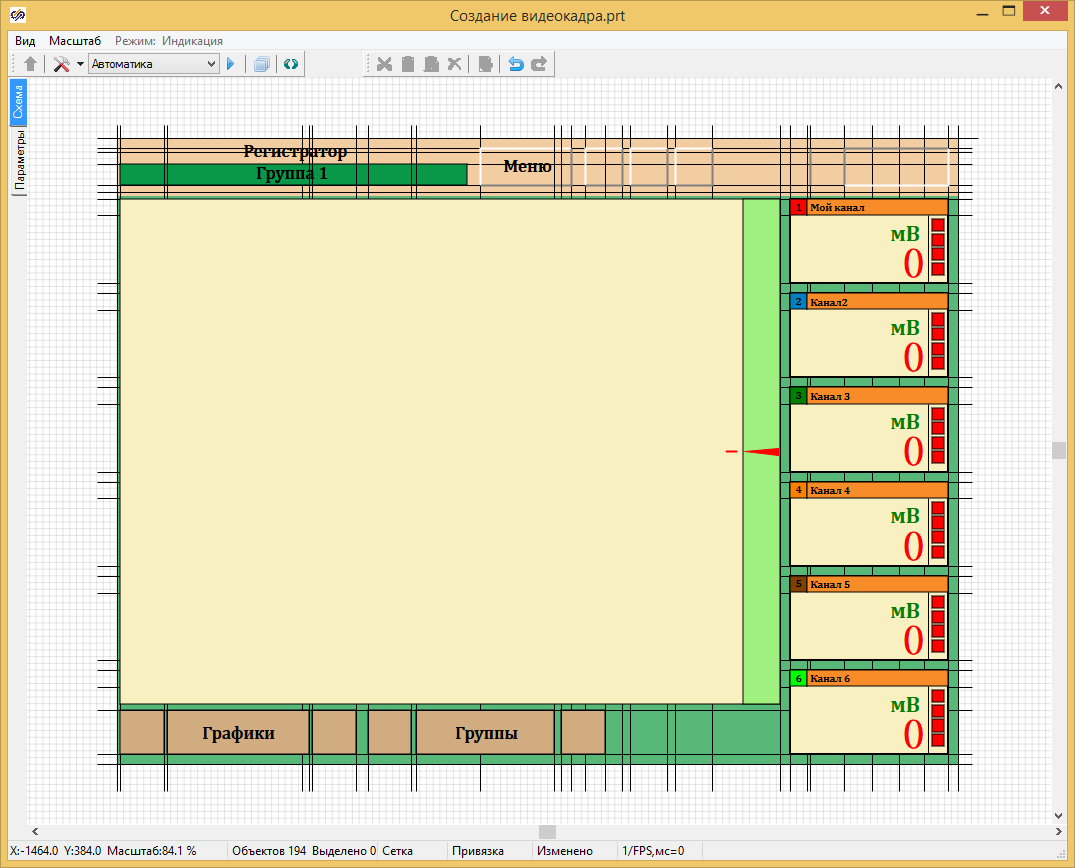


Рисунок 6. Внешний вид видеокадра после добавления уникальных блоков

1. Убираем на видеокадре все лишнее, добавляем необходимые детали (см. **Рисунок 7**).

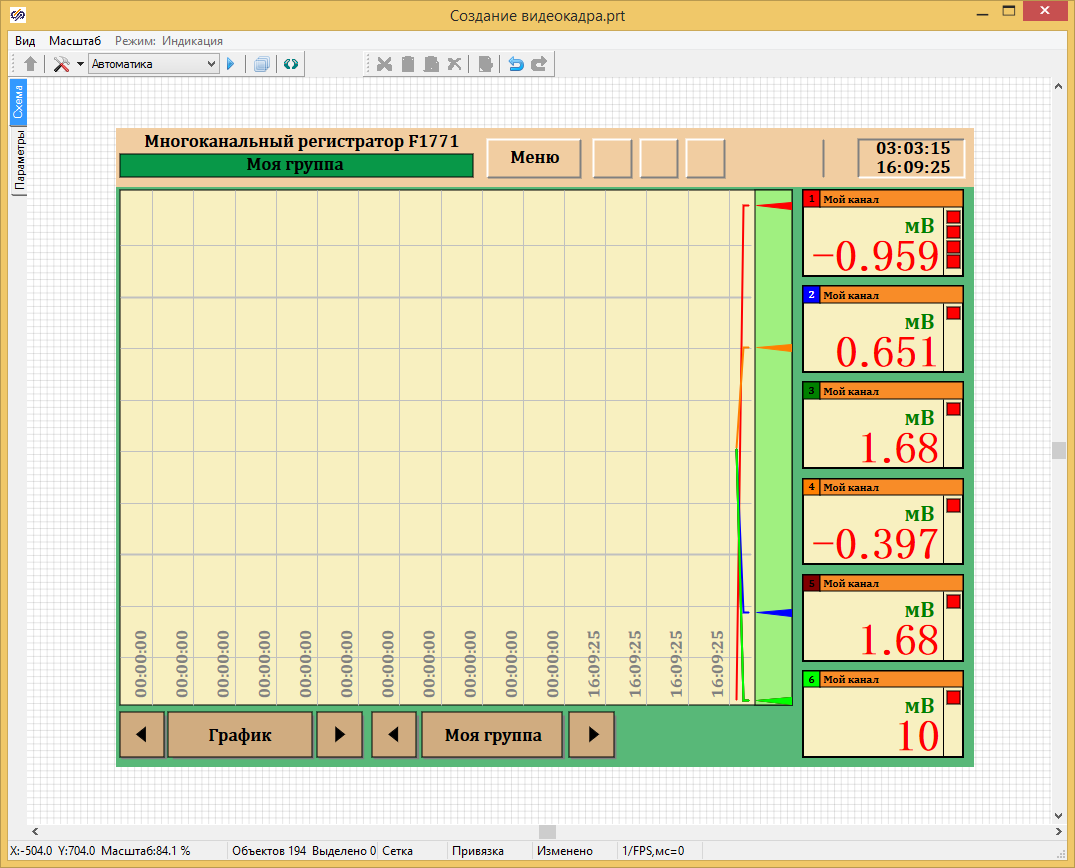


Рисунок 7. Финальный вид видеокадра перед окончательной проверкой

1. Производим окончательное тестирование блоков, визуальное отображение примитивов, скрипты. При соответствии исходному заданию создание видеокадра можно считать завершенным.