

## **РАЗРАБОТКА ВЕКТОРНОГО ГРАФИЧЕСКОГО РЕДАКТОРА НА БАЗЕ БИБЛИОТЕКИ QT В РАМКАХ ОТКРЫТОГО ПРОЕКТА OpenSCADA**

**Введение.** SCADA системы - это комплекс программных средств, применяющихся при создании автоматизированных систем управления (АСУ) промышленными объектами. В настоящее время основным элементом АСУ являются программируемые логические контроллеры, применение которых минимизирует участие человека в управлении техническими комплексами. Однако необходимость мониторинга и технологического процесса грамотными операторами-технологами остается. Для этого с помощью SCADA систем осуществляется связь контроллера с компьютером и предоставляется так называемый человеко-машинный интерфейс (HMI). При этом специалистам недостаточно получать численные значения тех или иных характеристик объекта (температуры, давления, расхода и т. д.). Опыт показывает, что наиболее информативной формой представления технологических процессов являются мнемосхемы - совокупность сигнальных устройств и сигнальных изображений оборудования и внутренних связей контролируемого объекта, выполняемые на персональном компьютере. Для их создания можно использовать любой из существующих графических редакторов. Однако полученные таким образом мнемосхемы являются статическими и не отражают динамику изменения характеристик процесса, а следовательно, они неадекватны и неудобны для восприятия. Таким образом, одной из задач, стоящих перед разработчиками SCADA систем, является создание графического редактора для изображения объектов, характеристики которых могут быть динамически изменены.

**Принципы и функции разработанного графического редактора.** Авторами статьи был разработан векторный графический редактор на базе библиотеки Qt в рамках открытого проекта OpenSCADA.

Основой описываемого редактора являются три графических примитива: линия, дуга, кривая Безье. К динамически изменяющимся характеристикам этих примитивов относятся:

1. Координаты контрольных точек: используются для задания формы линии, дуги или кривой Безье. При этом линия имеет 2 контрольные точки (рис. 1), дуга - 5 контрольных точек (рис.2), кривая Безье - 4 (рис.3)..
2. Ширина линии.
3. Цвет линии.
4. Ширина бордюра.
5. Цвет бордюра.
6. Стил линии (сплошная, пунктирная, точечная).

Примеры примитивов различного цвета, толщины, стилей с бордюрами и без бордюров приведены на рис.4.



Рисунок 1 - Контрольные точки линии

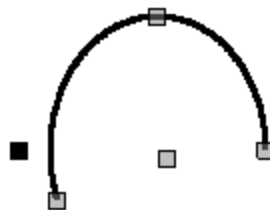


Рисунок 2 - Контрольные точки дуги

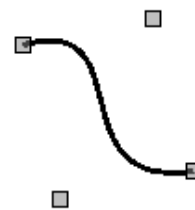


Рисунок 3 - Контрольные точки кривой Безье

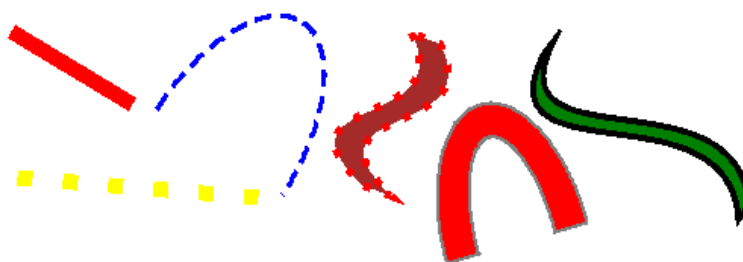


Рисунок 4 - Примеры примитивов различного цвета, толщины, стилей с бордюрами и без бордюров

Кроме того, разработчиками предусмотрена возможность связи различных графических примитивов для создания сложных графических объектов. Если связанные примитивы образуют замкнутый контур, то он может быть залит цветом и/или изображением (рис.5).



Созданный графический редактор позволяет масштабировать и поворачивать фигуры (примитивы и сложные графические объекты) (рис.6, рис.7).

К возможностям редактора относятся выделение, перемещение, копирование и удаление фигур.

Рисунок 5 — Заливки замкнутого контура цветом и изображениями

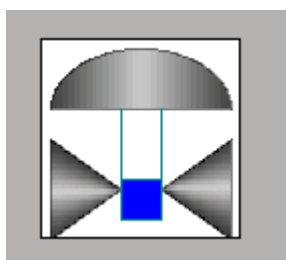


Рисунок 6 — фигура с масштабом 1 по ОХ и ОУ и нулевым углом поворота

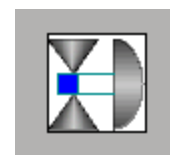





Рисунок 7 — фигура с масштабом 0,5 по ОХ и ОУ и углом поворота 90.

**Основные принципы работы в описываемом графическом редакторе.** Для того, чтобы приступить к работе с рассматриваемым графическим редактором после

запуска проекта OpenSCADA с помощью кнопки  необходимо вызвать «Рабочий пользовательский интерфейс». При этом откроется окно, состоящее из меню, панели инструментов и окон «Виджет» и «Атрибуты», находящихся в левой части экрана. (рис. 8).

В окне «Виджет» содержится перечень существующих графических библиотек и принадлежащих им элементов.

Предположим, что нам необходимо добавить графический элемент в одну из существующих библиотек. Для этого выделим имя библиотеки и нажмем кнопку  на панели инструментов. В появившееся окно введем идентификатор и имя нового графического элемента (рис. 9). После чего перейдем к этому элементу и нажмем кнопку . При этом справа появится поле для рисования. С помощью двойного щелчка мышью по этому полю перейдем в режим редактирования — режим работы графического редактора, позволяющий осуществлять все предусмотренные манипуляции с фигурами. В тот момент, когда вновь созданный нами графический элемент находится в фокусе, окно «Атрибуты» приобретет вид, представленный на рис.10.

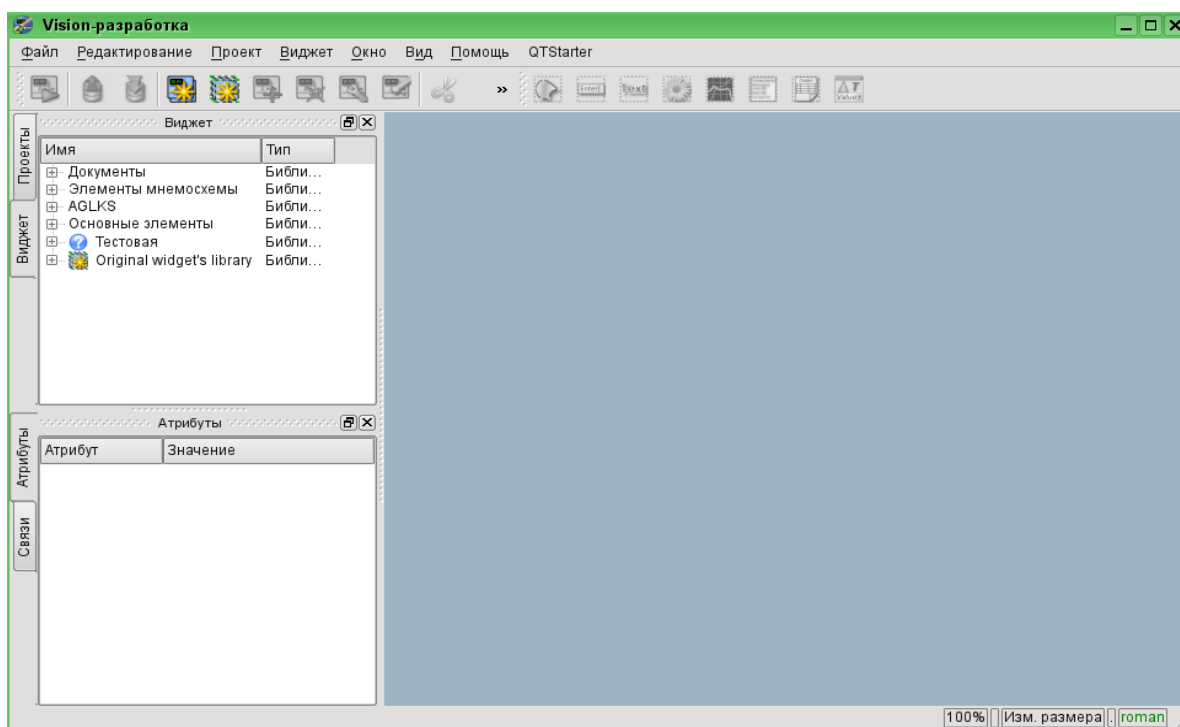


Рисунок 8 — Окно «Рабочего пользовательского интерфейса»

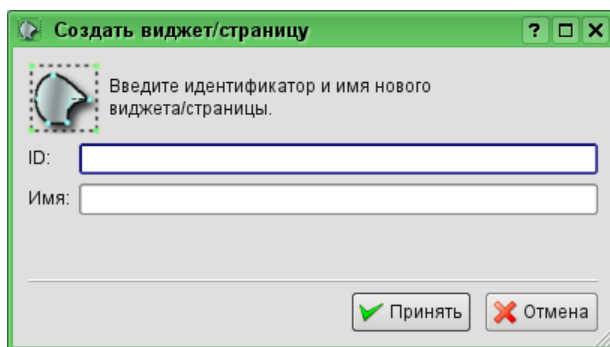


Рисунок 9 — Окно для создания нового графического объекта

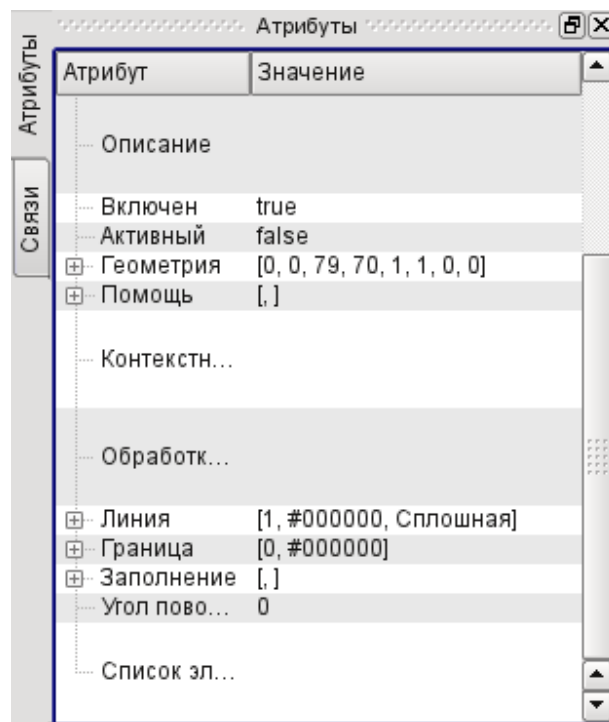


Рисунок 10 — Окно «Атрибуты»

С помощью мыши или поля «Геометрия» окна «Атрибуты» зададим размеры поля для рисования и масштабные коэффициенты.

Воспользовавшись полем «Линия» окна «Атрибуты» зададим ширину, цвет, стиль линий, которыми будем рисовать. С помощью поля «Граница» зададим ширину и цвет бордюра. Поле «Заполнение» позволяет задать цвет и изображение заливки. «Список элементов» содержит перечень примитивов и заливок, используемых при создании графического объекта.

Графические примитивы могут быть нарисованы с помощью мышки или заданы с помощью списка графических примитивов. В первом случае координаты контрольных точек создаваемого примитива вычисляются автоматически, ширина, цвет, стиль линии, ширина и цвет бордюра устанавливаются по умолчанию из окна «Атрибуты». Во втором случае примитив необходимо описать в «Списке элементов» следующим образом:

line:(x1|y1):(x2|y2):width:color:border\_width:border\_color:style (1)  
 arc:(x1|y1):(x2|y2):(x3|y3):(x4|y4):(x5|y5):width:color:border\_width:border\_color:style (2)  
 bezier:(x1|y1):(x2|y2):(x3|y3):(x4|y4):width:color:border\_width:border\_color:style, (3)

где (x1|y1) - координаты первой контрольной точки примитива,  
 (x2|y2) - координаты второй контрольной точки примитива,  
 (x3|y3) - координаты третьей контрольной точки примитива,  
 (x4|y4) - координаты четвертой контрольной точки примитива,  
 (x5|y5) - координаты пятой контрольной точки примитива,  
 width - ширина линии, которой будет нарисован примитив,  
 color - цвет линии, которой будет нарисован примитив,  
 border\_width - ширина бордюра,  
 border\_color - цвет бордюра,  
 style - стиль линии.

При этом приоритетными являются значения ширины, цвета, стиля линии, ширины и цвета бордюра, заданные в «Списке элементов». Если необходимо использовать какие-либо характеристики, устанавливаемые полями «Линия» или «Граница» окна «Атрибуты», то при описании примитива их следует пропустить. Например, мы хотим создать линию с шириной 3 красного цвета сплошную и без бордюра. При этом в поле «Линия» заданы ширина линии, равная 3, черный цвет линии, сплошной стиль линии; в поле «Граница» заданы ширина бордюра, равная 5, цвет бордюра зеленый. Тогда описание примитива в «Списке элементов» должно выглядеть следующим образом:

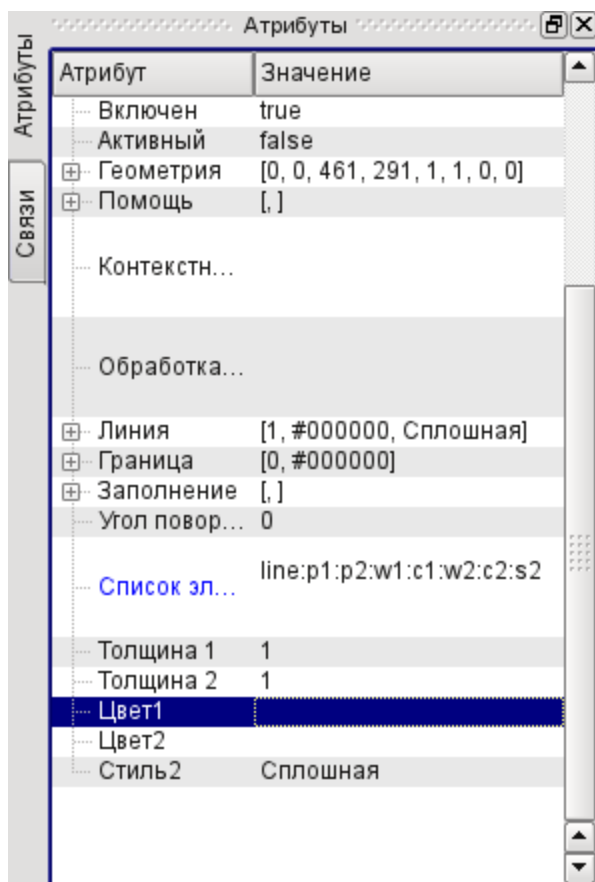
line:(x1|y1):(x2|y2)::red:0:: (4)

Формы записи (1) - (4) задают статические характеристики примитивов, которые не могут быть изменены программно. Для задания динамических характеристик необходимо использовать форму записи:

line:p1:p2:w1:c1:w2:c2:s1. (5)

Тогда в окно «Атрибуты» добавятся поля: Точка1 (p1), Точка2 (p2), Ширина1 (w1), Ширина2 (w2), Цвет1(c1), Цвет2(c2), Стили1(s1) (рис.11) Значения этих полей можно изменять программно, с помощью одного из языков программирования проекта OpenSCADA. Очевидно, что при желании динамическими могут быть объявлены не все свойства примитива, а одно или несколько, кроме того одно динамическое свойство может быть использовано многократно.

Редактирование координат контрольных точек осуществляется либо с помощью мыши, либо путем их изменения в «Списке элементов» (в статике) или окне «Атрибуты» (в динамике). Редактирование других характеристик осуществляется с помощью изменения содержимого полей «Линия», «Граница» окна «Атрибуты» или «Списке элементов».



Предусмотрены возможности выделения (левая кнопка мыши), совместного выделения примитивов (левая кнопка мыши + Ctrl), их перемещения (клавиатура/мышь), удаления (Del).

Для связи примитивов друг с другом необходимо:

1. Нажать кнопку Connections .
2. Выделить одну из фигур и переместить её начальную или конечную точку к желаемой начальной или конечной точке другой фигуры.

Для заливки замкнутого контура необходимо 2 раза кликнуть мышкой внутри него либо задать заливку в «Списке элементов» следующим образом:

fill:(x1:y1):(x2:y2):(x3:y3):color:image (статика),

fill:p1:p2:p3:c1:i1 (динамика),

где (x1:y1):(x2:y2):(x3:y3) - координаты начальных/конечных точек примитивов, образующих замкнутый контур,

Рисунок 11 — Окно «Атрибуты» с динамическими характеристиками объектов

образующих замкнутый контур,  
color - цвет заливки,  
image - изображение заливки.

При этом приоритетными являются значения цвета и изображения заливки, заданные в «Списке элементов». Если необходимо использовать характеристики, устанавливаемые полем «Заполнение» окна «Атрибуты», то при описании примитива их следует пропустить.

Для масштабирования фигур необходимо в поле «Геометрия» задать значения масштаба по осям X и Y. Кроме того, существует возможность визуально увеличить/уменьшить изображение, не меняя масштаб. Для этого необходимо выйти из режима редактирования, щелкнуть правой кнопкой мыши по свободному месту поля рисования, в появившемся контекстном меню выбрать пункт «Увеличить (+10%)»/«Уменьшить (-10%)» либо прокрутить колесико мышки, удерживая при этом CTRL. Выход из режима редактирования осуществляется либо с помощью нажатия клавиши Esc, либо с помощью меню, которое появляется в результате клика правой кнопкой мыши по свободному месту поля рисования.

Для поворота созданных объектов следует в окне «Атрибуты» задать «Угол поворота» от -360 до 360.

Выделение фигур осуществляется с помощью щелчка по ним левой кнопкой мыши или области выделения. Предусмотрено совместное выделение фигур с помощью клавиши Ctrl.

Перемещать фигуры можно как с помощью мыши, так и с помощью управляющих клавиш.

Копирование и удаление фигур выполняется с помощью Ctrl+C и Del соответственно.

Необходимо отметить, что разработанный авторами векторный графический редактор поддерживает прозрачность цвета, задаваемую следующим образом:

color-t,

где color - цвет,

t - прозрачность от 0 (полностью прозрачный) до 255 (непрозрачный).

Немаловажной является возможность сохранения созданных графических объектов во многих графических форматах. Для этого необходимо выйти из режима редактирования и в контекстном меню, открываемом при щелчке правой кнопкой мыши по пустому месту поля рисования, выбрать пункт «Создать изображение из виджета». При этом появится окно, позволяющее указать имя изображения и директорию, в которой оно будет сохранено.

**Выводы.** Таким образом, созданный векторный редактор выделяется достаточно высокой функциональностью, простотой и удобством применения и отвечает требованиям, предъявляемым к графическим редакторам, используемым в SCADA системах.