

УДК 528.835:681.5

**ПРОГРАММНАЯ КОМПОНЕНТА ВИЗУАЛИЗАЦИИ ДАТЧИКОВ  
ДЛЯ АВТОМАТИЗИРОВАННЫХ СИСТЕМ УПРАВЛЕНИЯ  
ЭЛЕКТРОННО-ЛУЧЕВЫМИ ЭНЕРГОКОМПЛЕКСАМИ**

*канд. техн. наук, доц. В.Г. ЗАЛЕССКИЙ, А.К. КОРОЛЬКОВ, П.В. ПАВЛОВЕЦ  
(Полоцкий государственный университет)*

*Представлены результаты разработки компоненты визуализации датчиков, представляющей собой элемент графического интерфейса для программ автоматизированных систем управления электронно-лучевыми энергокомплексами. Рассмотрены некоторые обобщенные принципы построения интерфейсов программ автоматизированных систем управления электронно-лучевыми энергокомплексами, состоящими из разнообразных устройств. Представлен пример использования разработанной компоненты визуализации датчиков для создания единообразного интерфейса пользователя автоматизированной системы управления электронно-лучевым энергокомплексом. Использование данной компоненты визуализации предназначено для повышения надежности и эффективности работы оператора с программой автоматизированной системы управления электронно-лучевым энергокомплексом. Причем компонента визуализации реализована в виде управляющего элемента ActiveX, что делает возможным ее применение для разработки программного обеспечения в любой среде программирования, поддерживающей технологию ActiveX.*

**Введение.** В настоящее время автоматизация является быстро развивающейся областью. Существует огромное количество различных предприятий и фирм, занимающихся как изготовлением оборудования и программного обеспечения для автоматизации, так и непосредственно самой автоматизацией. На сегодняшний день автоматизированные системы управления находят применение в различных отраслях народного хозяйства и область их применения неуклонно расширяется. Это связано в первую очередь с тем, что внедрение автоматизированных систем управления в технологические процессы повышает их эффективность. Это особенно актуально для современных технологических процессов, использующих сложное наукоемкое оборудование. Примером таких процессов может служить электронно-лучевая сварка с использованием электронно-лучевых энергокомплексов. Оснащение такого оборудования автоматизированными системами управления позволяет существенно повысить как производительность, так и комфортность работы. Однако автоматизированные системы управления по существу являются человеко-машинными системами. Человек-оператор является замыкающим звеном таких систем управления, следовательно, надежность и качество работы автоматизированных систем управления во многом зависят от оператора и его способности быстро и качественно воспринимать и обрабатывать информацию о контролируемых параметрах [1]. Использование графического интерфейса в программах управления автоматизированными системами повышает удобство восприятия и скорость обработки информации. Следовательно, задача проектирования и разработки интерфейса пользователя для программ автоматизированных систем управления (АСУ) имеет немаловажное значение. В работе представлены результаты разработки ActiveX компоненты, представляющей собой элемент графического интерфейса для программ АСУ электронно-лучевых энергокомплексов.

**Особенности построения интерфейса программ АСУ электронно-лучевых энергокомплексов**

Электронно-лучевые энергокомплексы несмотря на их огромное разнообразие всегда имеют общую структурную организацию: электронно-лучевая пушка, система позиционирования детали и вакуумная система. При этом для каждой такой подсистемы уже, как правило, существуют свои обособленные решения для построения визуальных интерфейсов с использованием графических компонент визуализации показаний датчиков для программ АСУ. Поэтому создание единой унифицированной компоненты визуализации показаний датчиков для всех систем обуславливает появление новых требований к ее функциональности:

- графический интерфейс, построенный с использованием этой компоненты должен обеспечивать возможность единообразно отображать различные датчики и позволять задавать для них диапазон измеряемых значений без каких-либо ограничений. Это требование обуславливается тем, что контролируемые параметры для электронно-лучевых комплексов весьма разнородны и измеряемые величины могут изменяться в широких диапазонах;

- необходимо предусмотреть возможность контроля выхода измеряемых значений из допустимой области и выдачу сигнала об этом событии оператору. При этом следует учесть, что отдельные датчики приобретают функциональность только при достижении некоторых пороговых значений одной из измеряемых величин;

- некоторые контролируемые параметры имеют очень маленькую область допустимых значений по сравнению со всем диапазоном измерения, например, давление в камере вакуумной системы. Это означает, что измеряемые параметры должны отображаться с определенной точностью вне зависимости от шкалы датчика с возможностью при необходимости изменения цены деления шкалы. Причем для некоторых контролируемых величин важным является не их абсолютное значение, а соответствие этого значения ограниченному диапазону. Для других значений, кроме привычной стрелочной индикации состояния, необходимо отображать (дублировать) точное значение измеряемого параметра;

- разрабатываемый интерфейс должен быть удобен для оператора, в том числе и для людей, без навыков работы с современными программными системами. Поэтому графическая компонента визуализации показаний датчиков, напоминающая обычные стрелочные индикаторы измерительных приборов обеспечивает оператору возможность быстро и качественно воспринимать и обрабатывать информацию о контролируемых параметрах.

Технология, применяемая для создания интерфейса, должна обеспечить создание приложений, собираемых из готовых компонентов. Причем эти компоненты не должны зависеть от того, на каком языке написаны как готовые компоненты, так и использующее их приложение. Исходя из этих требований, компонента визуализации датчиков для автоматизированных систем управления реализована в виде элемента управления ActiveX. Причины выбора этой технологии обусловлены тем, что, используя технологию ActiveX, программист по существу создает некоторый высокоуровневый, пригодный для многократного использования объект с конкретной полезной функцией [2]. Элемент управления ActiveX обладает уникальным идентификатором GUID и должен быть зарегистрирован в реестре. Это означает, что, создав элемент управления ActiveX, обладающий интересующей функциональностью, можно в дальнейшем позволить его пользователям встраивать этот элемент в свои приложения, написанные на различных языках программирования, при этом нет никакой необходимости предоставлять исходный код этой компоненты [3]. Таким образом, реализация компоненты визуализации датчиков для автоматизированных систем управления в виде элемента управления ActiveX обладает существенным рядом достоинств как с точки зрения сокрытия исходного кода компоненты, так и с точки зрения возможности применения компоненты при разработке другого программного обеспечения (в частности, программ автоматизированных систем управления).

#### **Компонента визуализации**

Компонента визуализации разрабатывалась с учетом вышеописанных требований к функциональности и на основе концепций объектно-ориентированного программирования. Диаграмма классов разработанной компоненты визуализации представлена на рисунке 1.

Классы компоненты визуализации были реализованы в соответствии с разработанной диаграммой классов (см. рис. 1). Ниже описана основная функциональность классов, рассмотрены их атрибуты и методы.

**CIndicator:** хранит данные о внешнем виде шкалы, содержит функции рисования шкалы, которые затем вызываются в методе OnDraw класса элемента управления. Наиболее важные методы класса CIndicator:

- деструктор ~CIndicator уничтожает созданные объекты GDI;
- метод Alarm управляет системой оповещения об аварийном состоянии;
- конструктор CIndicator производит инициализацию свойств класса, определяющих внешний вид компоненты, производит предварительные расчеты для последующей визуализации шкалы;
- CreateBrushes создает различные объекты GDI, необходимые для рисования компоненты;
- метод SetColors управляет цветами компоненты, определяет цветовую схему индикатора;
- метод Arch рисует дугу;
- ColorizeScale закрашивает зоны шкалы;
- SetArrow рисует указатель;
- SetCaptions рисует заголовки;
- SetCenter устанавливает центр окружностей, образующих шкалу;
- SetChord устанавливает размер хорды, соединяющей концы верхней дуги;
- SetDash1 рисует маленькие штрихи с интервалом dwDashInterval1;
- SetDash2 рисует большие штрихи с интервалом dwDashInterval2;
- SetRadius вычисляет радиус верхней дуги шкалы для конкретного окна;
- SetScale рисует подписи над делениями шкалы;
- методы Arch, ColorizeScale, SetArrow, SetCaptions, SetCenter, SetChord, SetDash1, SetDash2, SetRadius, SetScale служат для непосредственного рисования индикатора.

**CIndicatorApp:** класс приложения, генерируется при помощи средства MFC ActiveX ControlWizard.

**CIndicatorCtrl:** базовый класс элемента управления наследуется от класса COleControl – стандартного базового класса библиотеки MFC, который используется элементами управления OLE. Генерируется при создании проекта. Класс CIndicatorCtrl переопределяет некоторые методы базового класса: OnDraw, DoPropExchange, OnResetState.

**CIndicatorPropPage:** класс страницы свойств. Этот класс сгенерирован при создании проекта. Он реализует одну пустую страницу свойств. Этот класс является прямым потомком класса COlePropertyPage, базового класса всех страниц свойств.

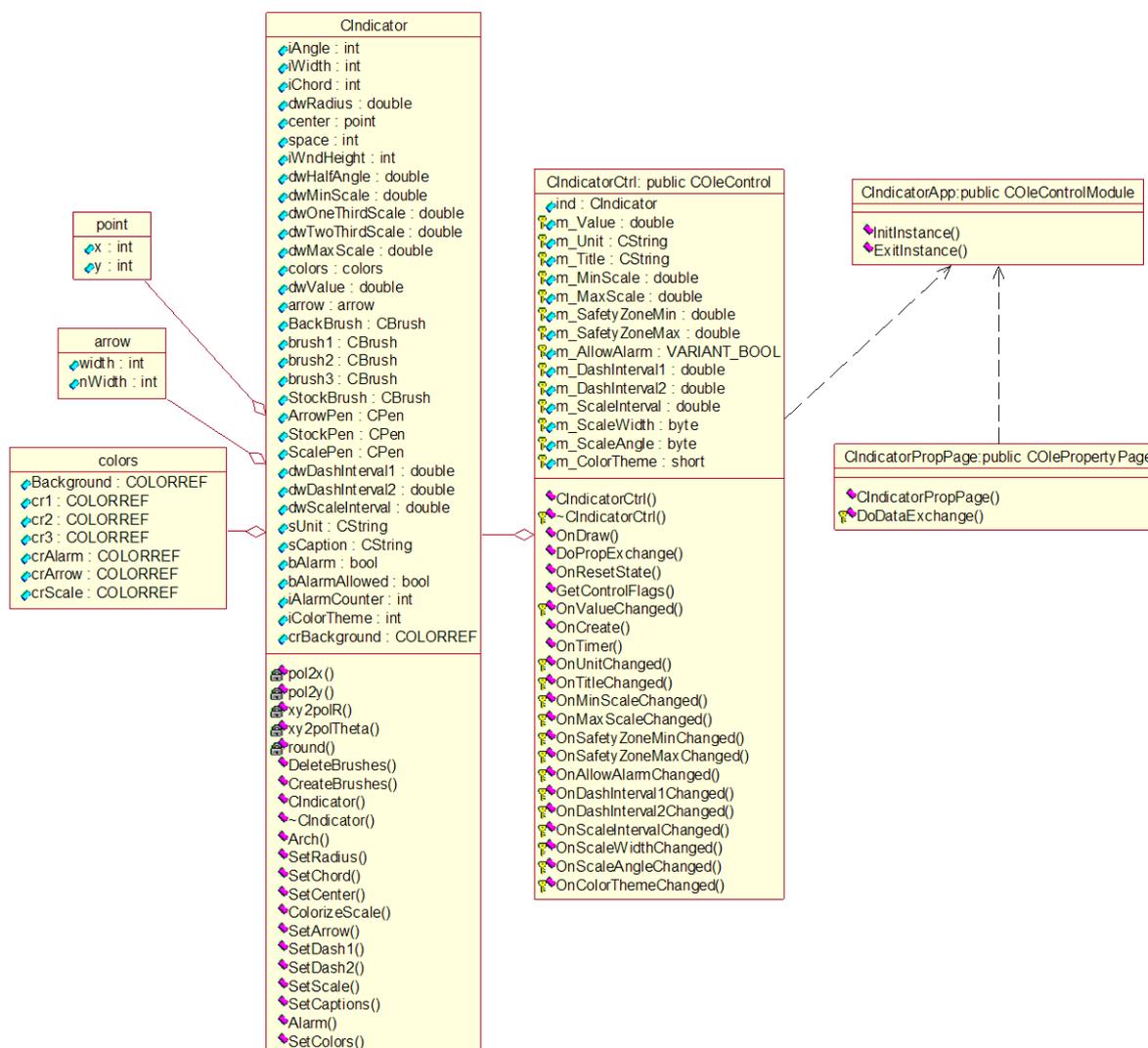


Рис. 1. Диаграмма классов разработанной компоненты визуализации

### Пример реализации датчиков на основе разработанной компоненты визуализации

Созданная компонента визуализации для автоматизированных систем управления представляет собой элемент ActiveX и позволяет реализовывать различные датчики, которые отображают измеряемую величину в виде значений на виртуальной шкале, при этом программист имеет возможность назвать измеряемую величину и задать название единиц измерения этой величины. Для этого используются следующие методы:

- SetTitle(char) устанавливает заголовок шкалы (измеряемая величина);
- SetUnit(char) задает единицы измерения;
- SetValue(double) устанавливает отображаемое значение.

Разработанная компонента визуализации также позволяет регулировать диапазон измеряемых значений произвольным образом, предоставляя тем самым программисту, который будет её использовать, широкий спектр возможностей как для обеспечения необходимых диапазонов измерения величин, так и для достижения необходимой точности измерений. Для этого используются следующие методы:

- SetDashInterval1(double) устанавливает расстояние между малыми штрихами;
- SetDashInterval2(double) устанавливает расстояние между большими штрихами;
- SetScaleInterval(double) устанавливает расстояние между подписями делений шкалы;
- SetMaxScale(double) устанавливает максимальное значение шкалы;
- SetMinScale(double) устанавливает минимальное значение шкалы.

Также следует отметить, что позиционирование стрелки указателя на шкале не привязано к делениям шкалы, а осуществляется по номинальному значению измеряемого параметра (метод `SetValue(double)`). Это означает, что у программиста есть возможность отобразить номинальное значение параметра, например, 1585,32 мВ на шкале с ценой деления в 100 мВ. Для удобства оператора также предусмотрено дублирование информации о номинальном значении измеряемого параметра в текстовом виде внизу компоненты. Виртуальная шкала обновляется автоматически, это упрощает индикацию физических величин, изменяющихся динамически.

Следует также отметить, что разработанная компонента визуализации датчиков обладает системой контроля выхода измеряемых параметров из области допустимых значений и в случае выхода значения измеряемой величины из допустимой области созданный элемент ActiveX просигнализирует об этом сменой цвета фона датчика на аварийный цвет. Для задания пороговых значений и установки флага опасности используются следующие методы:

- `SetAllowAlarm(bool)/GetAllowAlarm(bool)` устанавливает/получает флаг разрешения аварийной сигнализации;
- `SetSafetyZoneMax(double)` устанавливает максимальное безопасное значение, при котором не срабатывает сигнализация;
- `SetSafetyZoneMin(double)` устанавливает минимальное безопасное значение, при котором не срабатывает сигнализация.

Предусмотрена возможность изменения внешнего вида датчиков по желанию программиста, использующего компоненту визуализации. Для этого используются следующие методы:

- `SetScaleAngle(int)` устанавливает угол разворота шкалы;
- `SetColorTheme(int)` выбирает цветовую схему из предустановленных;
- `SetScaleWidth(int)` устанавливает ширину шкалы;

Возможный внешний вид датчиков на основе разработанной компоненты визуализации представлен на рисунке 2.

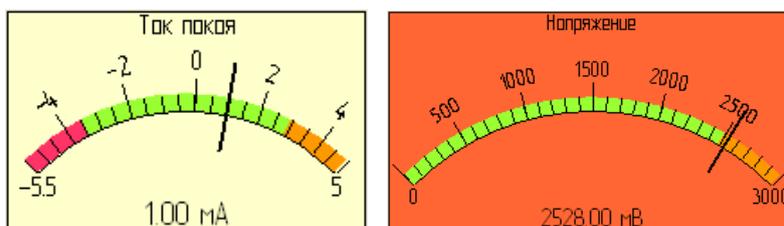


Рис. 2. Возможный внешний вид датчиков разработанной компоненты визуализации

**Заключение.** Рассмотрены некоторые обобщенные принципы построения интерфейсов программ АСУ энергокомплексов, состоящих из разнообразных устройств. Представлена разработанная компонента визуализации датчиков и пример ее использования для создания единообразного интерфейса пользователя. Использование данной компоненты предназначено для повышения надежности и эффективности работы оператора с программой автоматизированной системы управления. Причем компонента визуализации реализована в виде управляющего элемента ActiveX, что делает возможным ее применение для разработки программного обеспечения в любой среде программирования, поддерживающей технологию ActiveX.

#### ЛИТЕРАТУРА

1. Интерфейс пользователя с системой [Электронный ресурс]. – Режим доступа: [http://www.erudition.ru/referat/printref/id.35764\\_1.html](http://www.erudition.ru/referat/printref/id.35764_1.html). – Дата доступа: 19.11.2008.
2. ActiveX изнутри [Электронный ресурс]. – Режим доступа: <http://www.xserver.ru/computer/langprog/activex/4/>. – Дата доступа: 19.11.2008.
3. Создание элементов управления ActiveX [Электронный ресурс]. – Режим доступа: <http://www.xserver.ru/computer/langprog/activex/3/>. – Дата доступа: 19.11.2008.

Поступила 05.01.2009