Описание функционального блока Fan\_AVO

Функциональный блок служит для работы аппарата воздушного охлаждения (АВО).

**Fan\_AVOControl.cs**

Используемые библиотеки и классы

В данном разделе описываются использованные библиотеки и пространства имен, подключенные к функциональному блоку.

using ecl.commonClasses.BaseIm;

using ecl.commonClasses.BaseIm.ControlWnd;

using ecl.commonClasses.VentVs;

using ecl.StaticFbs.ProjectSettings;

using FB;

using System;

using System.ComponentModel;

using System.Drawing;

using System.Drawing.Drawing2D;

using System.Runtime.InteropServices;

using System.Windows.Forms;

using static ecl.commonClasses.WindowElements;

ecl.commonClasses.BaseIm – используется для вызова класса stdVisualPins;

ecl.commonClasses.BaseIm.ControlWnd – Используется для вызова интерфейса IPreviewPainter;

ecl.commonClasses.VentVs – Наследуется классом Fan\_AVOControl

ecl.StaticFbs.ProjectSettings – Используется для вызова класса ProjectSettings (класс с заданными переменными);

FB – это стандартная библиотека MasterScada:

System – Это стандартная библиотека C#;

System.ComponentModel – Используется для вызова класса DisplayNameAttribute, для отображения имени свойства;

System.Drawing – Предоставляет доступ к основным графическим функциям GDI+;

System.Drawing.Drawing2D – Расширяет функциональные возможности для создания двухмерной и векторной графики;

System.Runtime.InteropServices – Предоставляет разнообразные члены, поддерживающие COM-взаимодействие и службы вызова платформы;

System.Windows.Forms – Содержит классы для создания приложений Windows, которые позволяют наиболее эффективно использовать расширенные возможности пользовательского интерфейса, доступные в операционной системе Microsoft Windows;

static ecl.commonClasses.WindowElements – Используется для вызова статичного класса WindowElements;

Код

[ComVisible(true),

Serializable,

Guid("EBC455AE-9F1C-4E84-928F-407399920ED0"),

DisplayName("Вентилятор АВО")

]

[FBOptions(FBOptions.SupportObjectSateChanged)]

Serializable – Атрибут, определяющий метод сохранения ФБ в проекте MasterSCADA. Возможны два варианта: “Serializable” – для бинарного сохранения ФБ или “SerializationType(SerializationType.Xml)” – для сохранения настроек ФБ в текстовом виде в формате XML.

ComVisible(true) и Guid – Атрибуты для регистрации ФБ как COM объекта. Значение атрибута Guid должно быть уникальным;

DisplayName – Атрибуты для регистрации нашего класса как ФБ;

Класс GetConnectColor – устанавливает цвет отрисовки буквы при разрыве соединения с сервером

private Color GetConnectColor(){}

Перегруженный метод OnPaint из класса VentVsControl предназначен для отрисовки границ, сглаживания линий, заливки прозрачным фоном, задания ориентации (вертикальной или горизонтальной), отрисовки элементов поля, отрисовки текста: местный, дистанционный; автоматический, ручной.

protected override void OnPaint(PaintEventArgs e){}

Метод drawIndicator служит для описание графического элемента "Индикатор". Изменяет цвет "Индикатор" по состоянию, рассчитывает координаты точек "Индикатора" относительно размеров поля, рисует контур "Индикатора".

private void drawIndicator(Graphics g, int width, int height){}

Метод GetStatusColor возвращает статус цвета.

private Color GetStatusColor(){}

Метод drawBlades служит для описание графического элемента "Лопатки".

private void drawBlades(Graphics g, int width, int height){}

Перегруженный метод ToRuntime служит для перехода в Runtime.

protected override void ToRuntime(){}

Метод Commands\_AllCommandsDisableChanged служит для вызова ошибки (или метода) об недействительном предварительном просмотре если выполняются условия.

private void Commands\_AllCommandsDisableChanged(bool enabled){}

Перегруженный метод ToDesign служит для перехода в Designtime.

protected override void ToDesign(){}

Перегруженный метод OnVisibleChanged вызывает себя с аргументом visible и если перемнная wnd не равняется null то закрывает форму.

protected override void OnVisibleChanged(bool visible){}

Перегруженный метод OnMouseDown включается при нажатии кнопки

protected override void OnMouseDown(MouseEventArgs me){}

Метод Paint предназначен для работы с классом PaintEventArgs и отрисовки элементов полей с помощью класса Pen.

public void Paint(PaintEventArgs e, SizeF size){}

Метод GetAspectRastio служит для получения соотношения сторон.

public float GetAspectRastio(){}

**Fan\_AVO.cs**

Используемые библиотеки и классы

В данном разделе описываются использованные библиотеки и пространства имен, подключенные к функциональному блоку.

using ecl.commonClasses.BaseIm.ControlWnd;

using ecl.commonClasses.VentVs;

using ecl.visualControls.DiskretBox;

using FB;

using FB.VisualFB;

using InSAT.Library.Interop;

using System;

using System.ComponentModel;

using System.Runtime.InteropServices;

ecl.commonClasses.BaseIm.ControlWnd – Используется для вызова SpecialCommandDescriptor;

ecl.commonClasses.VentVs – Наследуется классом Fan\_AVO;

ecl.visualControls.DiskretBox – Используется для вызова класса DiscreteIndicatorControl;

FB – это стандартная библиотека MasterScada:

FB.VisualFB – Используется для вызова атрибута VisualControls;

InSAT.Library.Interop – Используется для вызова атрибутов CatID и CatIDs;

System – Это стандартная библиотека C#;

System.ComponentModel – Используется для вызова класса DisplayNameAttribute, для отображения имени свойства;

System.Runtime.InteropServices – Предоставляет разнообразные члены, поддерживающие COM-взаимодействие и службы вызова платформы;

Код

[Serializable,

ComVisible(true),

Guid("B84289A6-93A2-44DF-9EDB-374BC705C790"),

CatID(CatIDs.CATID\_OTHER),

DisplayName("Вентилятор АВО"),

VisualControls(typeof(Fan\_AVOControl), typeof(DiscreteIndicatorControl))

]

Serializable - атрибут, определяющий метод сохранения ФБ в проекте MasterSCADA, для бинарного сохранения ФБ.

ComVisible и Guid - атрибуты для регистрации ФБ как COM объекта. Значение атрибута Guid уникально.

CatID и DisplayName - атрибуты для регистрации нашего класса как ФБ. Атрибут CatID принимает перечисление CatIDs и задает категории ФБ в которых он появиться в Палитре (у ФБ может быть несколько атрибутов CatID). После обновления палитры ФБ он будет добавлен в категории, которые вы указали с помощью атрибутов CatID.

VisualControls(typeof(FlapControl), typeof(DiscreteIndicatorControl)) – атрибут (VisualConrols) для создания связи между ФБ и контролами.

Публичный класс Fan\_AVO наследуется от класса VentVs

public class Fan\_AVO : VentVs {}

В нем описывается перегруженный массив класса SpecialCommandDescriptor.

public override SpecialCommandDescriptor[] getSpecialCommands(){}

В котором создаются два объекта этого класса с тремя аргументами: «Открыть» new SpecialCommandDescriptor(Commands.CMDOPEN,"Включить",true), И «Закрыть» new SpecialCommandDescriptor(Commands.CMDCLOSE,"Отключить",true).

**Fan\_AVO.xml**

В этом файле находится список всех входов/выходов ФБ.

Структура элемента <Map> - описывает коллекции входов/выходов и групп. <VisualMap> - Описание входов выходов и для обмена данными с контролом.

У всех элементов кроме <Items> и <ItemsRange> не обходимо указывать уникальный ID. У элемента <Map>может быть только один из элементов <Items> или <ItemsRange>.

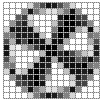
<Pin/> - описание входа ФБ, в его структуре описывается уникальный номер (ID), имя (Name) и тип (Type). <Pout/> - описание выхода ФБ, его структура идентична <Pin/>.

<Group> - описание группы ФБ, в данном случае это описание пользовательских выходов и пользовательских команд. Структура элемента <Group> полностью идентична структуре элемента <Map > , за исключением дополнительных атрибутов, таких как Name – имя группы и ID – уникальный номер группы.

Структура элемента <Events> …</ Events > содержит <Event> – описание обычного сообщения. Например: «<Event ID="1" Name="Команда" Category ="Information" Flags="DisableAck" />», где 1 (ID) – идентификатор сообщения, передается в функции FireEvent, SetEventState, «Команда» (Name) – текст сообщения по умолчанию, «Information» (Category) – это категория, «DisableAck» (Flags) – флаг, означающий что сообщение не требует квитирования. В данном случае используется только этот флаг. Остальные сообщения построены по такому же принципу (в некоторых сообщениях используется другая категория – Warning).

Структура элемента <Rights>…</Rights> описание действий, которые могут совершаться пользователем из окна управления или по горячим клавишам. SystemRight задает, что оператор может выполнять определенное системное действие над этим ФБ, в данном случае: FBChangeState - изменить состояние; FBOpenControlWindow – открыть окно управления; FBConfirm – квитировать.

**Fan\_AVO.bmp.**



Это изображение ФБ в формате bmp-файла.