

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Ильшат Ринатович Мухаметов

Должность: директор

Дата подписания: 13.07.2023 12:35:18

Уникальный программный ключ:

aba80b84033c9ef19b188e7ea0434f90a83a40954ba270e84b5c6e6402d1d8d0

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное бюджетное образовательное учреждение высшего образования «Казанский национальный исследовательский технический университет

им. А.Н. Туполева-КАИ»

(КНИТУ-КАИ)

Чистопольский филиал «Восток»

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНЫМ РАБОТАМ**

по дисциплине

**РАСПРЕДЕЛЕННЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ**

Индекс по учебному плану: **Б1.В.ДВ.02.03**

Направление подготовки: **09.03.01 Информатика и вычислительная техника**

Квалификация: **Бакалавр**

Профиль подготовки: **Вычислительные машины, комплексы, системы и сети**

Типы задач профессиональной деятельности: **проектный,**

**производственно-технологический**

Рекомендовано УМК ЧФ КНИТУ-КАИ

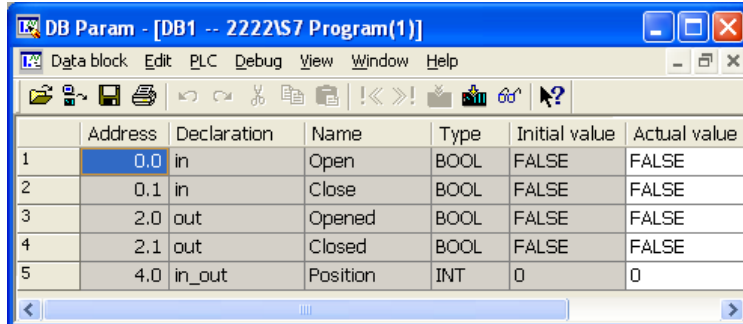
Чистополь

2023 г.

## Лабораторная работа № 1

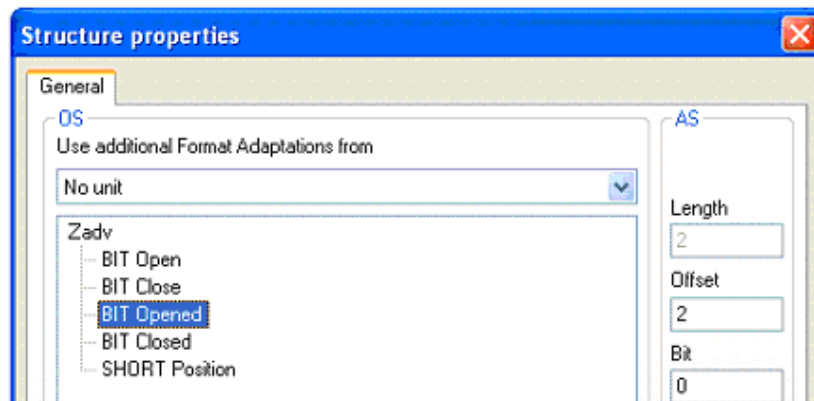
### Создание структурного тега для задвижки и дозатора

В Step 7 для управления задвижкой создан экземплярный блок данных DB\_Zadv. Ниже приведены поля этого экземплярного блока данных:



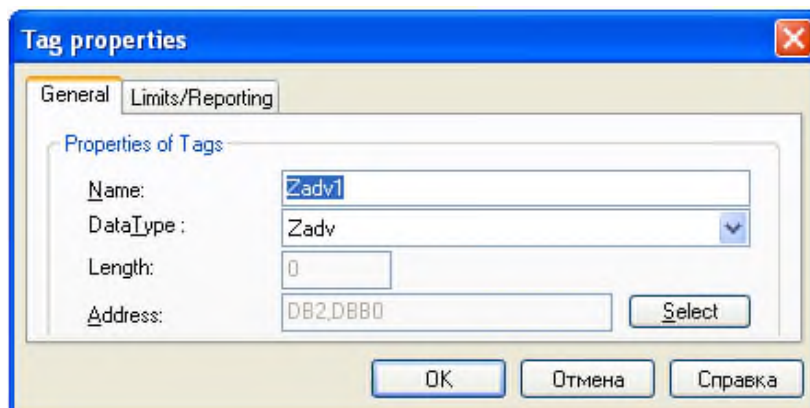
	Address	Declaration	Name	Type	Initial value	Actual value
1	0.0	in	Open	BOOL	FALSE	FALSE
2	0.1	in	Close	BOOL	FALSE	FALSE
3	2.0	out	Opened	BOOL	FALSE	FALSE
4	2.1	out	Closed	BOOL	FALSE	FALSE
5	4.0	in_out	Position	INT	0	0

В проекте WinCC создайте структурный тег Zadv. Ниже указано окно создания структурного тега.

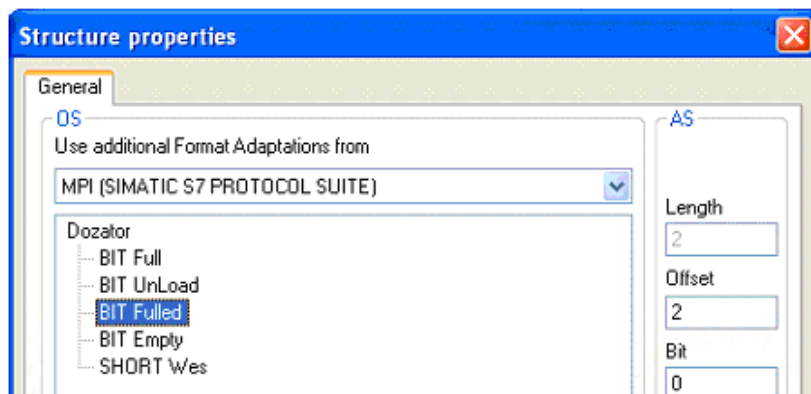


Для совместимости экземплярного блока данных DB\_Zadv и структурного типа Zadv сравните адресов полей, чтобы эти адреса были одинаковыми.

На основе этого структурного тега создайте тегов процесса для задвижки.

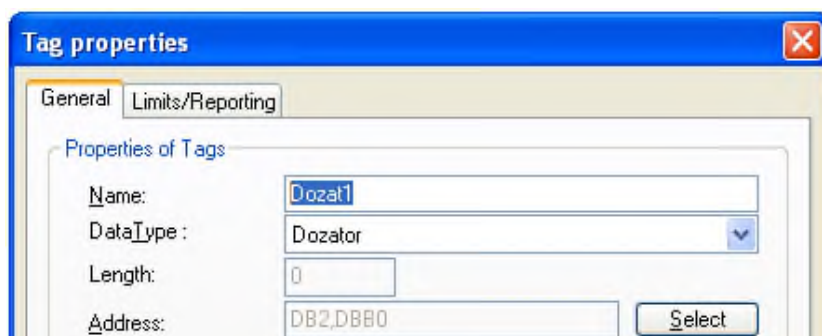


Дозаторы предназначены для набора определенного количества материалов. В проекте Step 7 был создан экземплярный блок данных DB\_Dozat. В проекте WinCC создайте структурный тег Dozator совместимый с этим блоком данных.



Напомним, что адреса некоторых полей структурного тега Dozator придется переопределить, так как поля Filled и Empty являются полями типа выход и у них адреса начинаются с нового слова.

На основе этого структурного тега создайте тег процесса Dozat1.

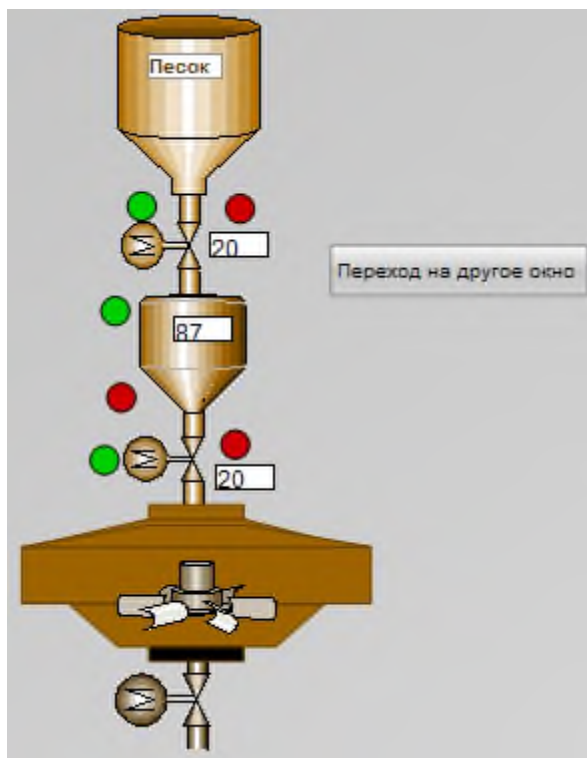


В конце отметим, что теги Zadv1 и Dozat1 будут использованы в дальнейших разработках экранной формы и динамизации объектов экранной формы. Поэтому создание этих тегов обязательно, с точки зрения дальнейшего изучения WinCC.

### Создание экрана процесса для БСУ

Запускайте Редактор Graphics Designer и откройте Палитру объектов. Внимательно изучайте инструментов Палитры объектов. Также откройте библиотеку рисунков и просмотрите рисунков из различных разделов этой библиотеки.

Нарисуйте объектов фрагмент экрана процесса для бетоносмесительной установки или для технологического проекта, согласно варианта выполнения курсового проекта. Экран процесса должен максимально отображать возможностей технологического оборудования.



Размещайте датчиков и исполнительных устройств технологического оборудования. В качестве примеров датчиков и исполнительных устройств используйте объектов экранов процесса реальных проектов.

## Лабораторная работа № 2

### Выбор главного экрана процесса и запуск проекта для БСУ

Определите экран процесса BSU.pdl как стартового окна при запуске проекта в режим Runtime. На этом экране процесса разместите кнопку для переключения на другой главный экран процесса. Напишите VB-скрипт для обработчика события мыши этой кнопки, который переключает на другое главное окно, например, на экран процесса Screen.pdl.

```
Sub OnClick(ByVal Item)
```

```
HMIRuntime.BaseScreenName = "Screen"
```

```
End Sub
```

Создайте другой главный экран процесса, разместите на него одну единственную кнопку. В обработчике события мыши этой кнопки напишите VB-скрипт для обратного переключения на главный экран процесса бетоносмесительной установки.

```
Sub OnClick(ByVal Item)
```

```
HMIRuntime.BaseScreenName = "BSU"
```

```
End Sub
```

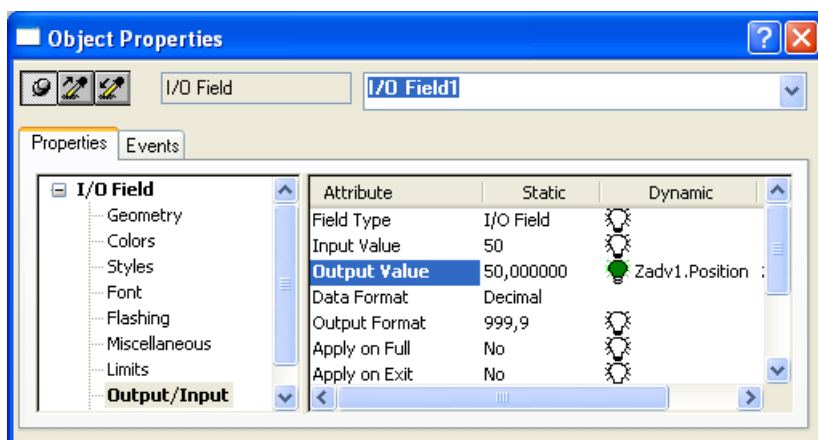
Настройте параметров запуска проекта WinCC. Запускайте проект в режим Runtime.

Измените экран процесса (добавьте какой-нибудь объект) и сохраняйте экран процесса. Не выходя из режима Runtime, нажимайте на кнопку перехода на другое окно и затем обратно переходите обратно в окно BSU.pdl. Обращайте внимание, что внесенные изменения на экран процесса BSU.pdl появляются.

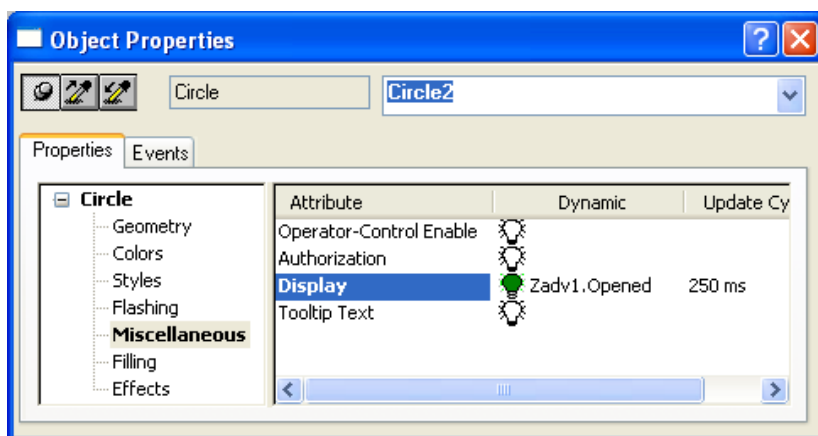
Тем самым, не выходя из режима Runtime можно внести изменения на экран процесса и посмотреть на результат работы этих изменений.

### Динамизация объектов рисунка главного окна

Определите динамику для датчика положения задвижки и датчика веса дозаторы с помощью числовых тегов. При этом, например, окно свойств рисунка датчика положения задвижки имеет вид:



Определите динамику свойства Display для дискретных датчиков задвижки и дозатора с помощью логических тегов. Например, окно свойств дискретного датчика задвижки «Открыта» имеет следующий вид:

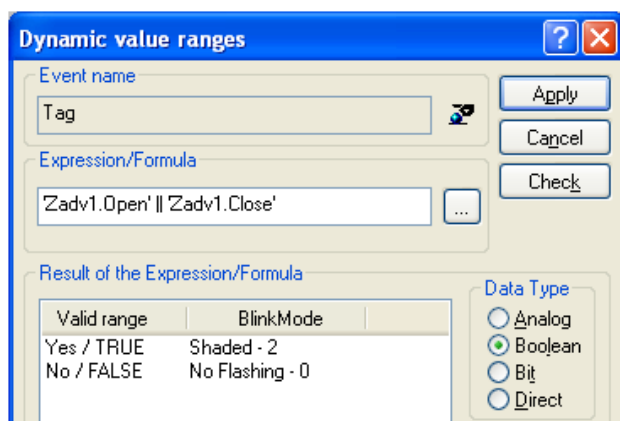


Запускайте проект Step 7 и WinCC одновременно в режим выполнения. В проекте Step 7 запускайте процесс приготовления порции бетона и следите за изменениями значений датчиков задвижки и дозатора.

### Лабораторная работа № 3

#### Динамизация объектов рисунка с помощью окна **Dynamic Dialog**.

При открытии/закрытии рисунок задвижки должна мигать. Для определения мигания рисунка задвижки необходимо определить динамику для свойства **BlinkMode**. Выделите это свойство и из контекстного меню выберите команду **Dynamic Dialog**. При этом появляется окно определения условия мигания.

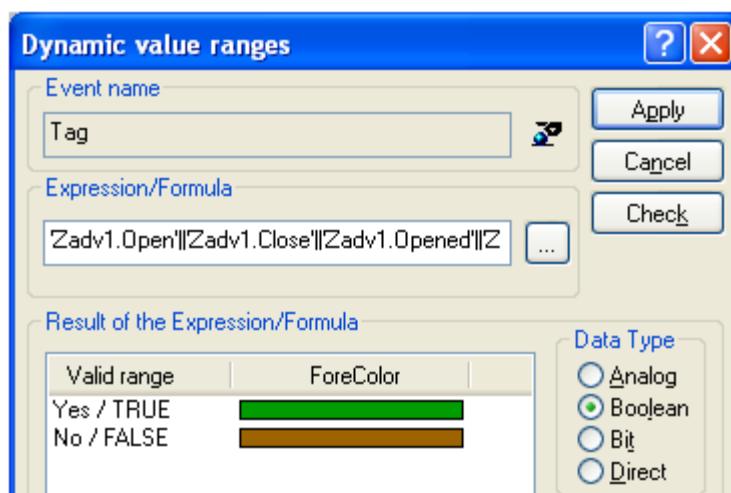


Как видно, здесь имеется 4 варианта определения выражения для динамизации. В этом рисунке выбран вариант **Boolean**, поэтому следует написать булево выражение, при выполнении которого, рисунок задвижки начинает мигать. Определите условие мигания задвижки в виде ИЛИ-условия:

**'Zadv1.Open' || 'Zadv1.Close'**

и режима мигания – **Shaded-2**.

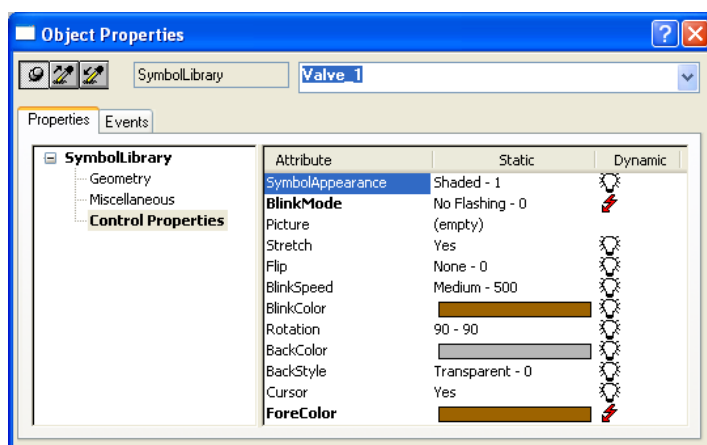
Для выбора фонового цвета необходимо определить динамику для свойства **ForeColor**. Для этого свойства также определите способ динамизации с помощью окна **Dynamic Dialog**.



Здесь условием зеленого фонового цвета является выражение:

**'Zadv1.Open' || 'Zadv1.Close' || 'Zadv1.Opened' || 'Zadv1.Closed'**

Для согласованной работы необходимо настроить и других свойств рисунка задвижки. Ниже приведено окно свойств рисунка задвижки.



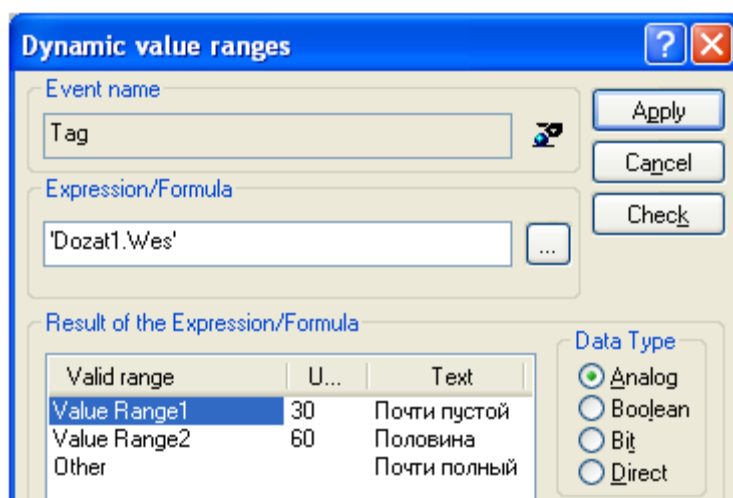
Как видно, свойства, у которых динамика определена с помощью Dynamic Dialog отмечены “молнией” красного цвета. Свойство SymbolAppearance имеет значение Shaded-1, в результате рисунок задвижки отображается плавным фоновым цветом свойства ForeColor.

Определите динамику для рисунка дозатора с помощью условия Analog. Для примера разместите на экранной форме объект типа Static Text, который предназначен для вывода текста об уровне заполнения дозатора:





Для свойства **Text** определите условие динамизации с помощью окна Dynamic Dialog.

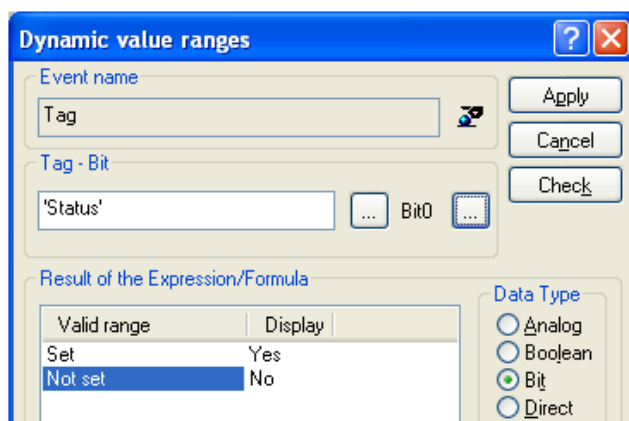


Как видно, в зависимости от диапазона значений датчика веса дозатора в текстовом поле отображаются различные тексты.

Для отображения режимов приготовления бетона на экране процесса BSU.pdl создайте информационную панель следующего вида:

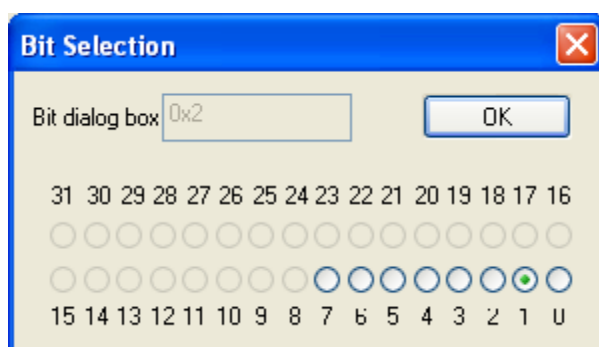


Индикаторы режима представлены в виде маленьких окружностей. Эти индикаторы должны отображаться только тогда, когда этот режим активен. Для определения такого способа динамизации, выберите свойство **Display** и для него создайте динамику с помощью окна Dynamic Dialog.



Как видно здесь в качестве тега выбран целочисленный тег Status. Предполагается, что вы раньше создали этот тег, и в качестве адреса этого тега выбрали маркерный байт MB0.

Для выбора бита этого тега нажмите на кнопку «многоточие». При этом появляется окно выбора бита, где следует выбрать нужный бит.



Таким образом, определите динамику для всех индикаторов режима и проверьте работу этих индикаторов в режиме исполнения WinCC.

Запускайте проект Step 7 и WinCC одновременно в режим выполнения. В проекте Step 7 запускайте процесс приготовления порции бетона и следите за изменениями фонового цвета и режима мигания рисунка задвижки.

## Лабораторная работа № 4

### Обработка события в режиме Direct Connection

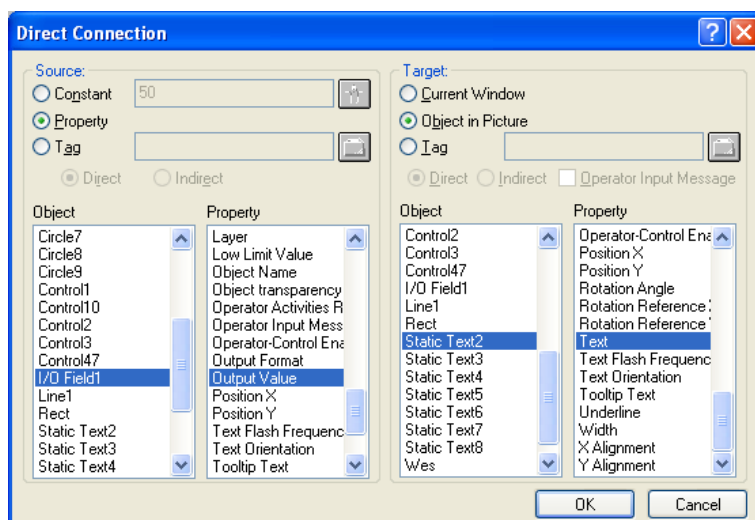
Режим Direct Connection предназначен для передачи данных на другой объект экрана процесса или для передачи данных на программу нижнего уровня через теги процесса. Таким образом, этот режим удобен для управления технологическим процессом из диспетчерского окна. Ниже приведем различные варианты использования окна Direct Connection.

Реализуйте передачу константы на тег процесса.

Для этого в левой части окна Direct Connection выберите переключатель Constant, а в правой части окна тег Dozat1.Wes:



Реализуйте передачу значения свойства текстового поля на свойство другого объекта. Например, значение текстового поля ввода/вывода передается на свойство Text объекта типа Static Text.



Реализуйте передачу значения свойства текстового поля тегу процесса. В этом случае в правой части окна Direct Connection следует выбрать тег процесса.

Запускайте проект Step 7 и WinCC одновременно в режим выполнения. В проекте Step 7 запускайте процесс приготовления порции бетона. Время от

времени в режиме Direct Connection передайте данных на теги и следите за изменениями в программе нижнего уровня.

### Синтаксис написания VB-скриптов

В той лабораторной работе на экране разместите различных кнопок и для них определите различных обработчиков событий мыши в виде VB-скрипта.

Использование параметра Item обработчика события мыши. Этот параметр определяет ссылку на объект, для которого написан этот обработчик. Для примера определите следующий VB-скрипт.

```
Sub OnClick(ByVal Item)
```

```
    MsgBox Item.ObjectName
```

```
    Item.ForeColor = RGB(255, 0, 0)
```

```
    Item.FontSize = 16
```

```
    Item.Text = "Моя кнопка"
```

```
End Sub
```

Коллекция `HMIRuntime.Screens` содержит ссылки на все экраны проекта, которые в данный момент открыты. К экранам можно обращаться по имени или по индексу экрана процесса. Например, получить ссылку на открытый экран `Signal.pdf` по имени можно так:

```
Set objScreen = HMIRuntime.Screens("Signal")
```

Использование индексов экрана позволяет просмотреть всех открытых экранов процесса. Для примера разместите на форме кнопку и определите для него обработчик события мыши в виде следующего VB-скрипта:

```
Sub OnClick(ByVal Item)
```

```
    Dim i, objScreen
```

```
    For i = 1 To HMIRuntime.Screens.Count
```

```
        Set objScreen = HMIRuntime.Screens(i)
```

```
MsgBox "Экран № "& i & " = " &  
objScreen.ObjectName
```

```
Next
```

```
End Sub
```

Объекты, размещенные на экранной форме доступны через коллекцию ScreenItems, экранной формы. К объектам также можно обращаться как по индексу, так и по имени объекта. Для примера разместите на форме кнопку и определите для него обработчик события мыши в виде следующего VB-скрипта:

```
Sub OnClick(ByVal Item)
```

```
Dim Screen, Obj, Wes
```

```
Set Screen = HMIRuntime.Screens("BSU")
```

```
Set Obj = Screen.ScreenItems("Wes")
```

```
Wes = Obj.OutputValue
```

```
MsgBox "Значение датчика веса = " & Wes
```

```
End Sub
```

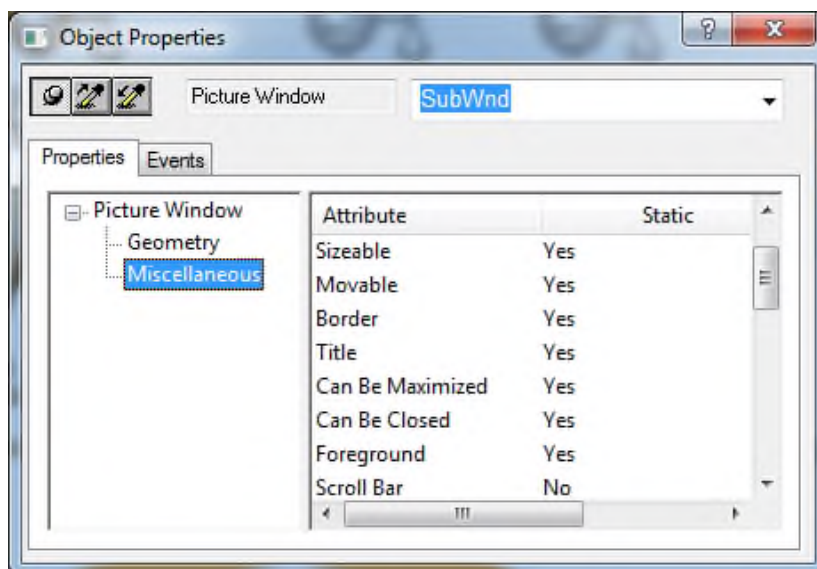
В этом примере происходит считывание свойства OutputValue текстового поля "Wes". К объектам главного экрана можно обращаться без ссылки на экран. Например, для получения ссылки на текстовое поле "Wes" главной формы достаточно написать:

```
Set Obj = ScreenItems("Wes")
```

В этом случае обработчик события мыши получается значительно короче.

Для отображения вспомогательного экрана в главном экране следует разместить объект типа **Picture Window** из Палитры объектов. После размещения на экране процесса этот объект примет вид прямоугольника, внутри которого будет отображаться вспомогательное окно.

Откройте окно свойств этого объекта и введите имя объекта SubWnd, чтобы обращаться к нему в коде скрипта. В окне свойств переходите в раздел Miscellaneous и установите характеристик окна (Sizeable, Moveable, Title и т.д) на значение Yes:



Предположим, что экран процесса Signal.pdl предназначен для отображения характеристик датчика веса и тренда изменения веса. Пусть это окно отображается при щелчке на текстовом поле датчика веса. Для этого откройте окно свойств текстового поля датчика веса и определите следующий обработчик события мыши:

```
Sub OnClick(ByVal Item)
```

```
Dim Obj
```

```
Set Obj = ScreenItems("SubWnd")
```

```
Obj.PictureName = "Signal"
```

```
Obj.CaptionText = "Датчик веса для Песка"
```

```
‘ Определение размера окна
```

```
Obj.Left = 570
```

```
Obj.Top = 80
```

```
Obj.Width = 600
```

```
Obj.Height = 200
```

```
End Sub
```

Как видно, в этом коде определяется имя экрана процесса, заголовок экрана и размеры экрана. Теперь сохраняйте экран процесса BSU.pdl и перезагружайте этот экран. При двойном щелчке на текстовом поле датчика веса появляется вспомогательное окно Signal.pdl.

Коллекция **Tags** содержит ссылок на всех тегов проекта. В отличие от других коллекций, к тегам *нельзя обращаться по индексу*, к тегам можно обращаться только по имени тега.

Для примера разместите на форме кнопку и определите для него обработчик события мыши в виде следующего VB-скрипта:

```
Sub OnClick(ByVal Item)
```

```
Dim Tag, str
```

```
Set Tag = HMIRuntime.Tags("Dozat1.Wes")
```

```
str = Tag.Read
```

```
MsgBox "Прочитанное значение Wes: " & str
```

```
Tag.Write "100"
```

```
MsgBox "Значение тега Wes: " & Tag.Value
```

```
End Sub
```

Как видно, в этом обработчике вначале считывается текущее значение тега Dozat1.Wes. Затем изменяется значение тега и еще раз прочитается.

Запускайте проект Step 7 и WinCC в режим выполнения. В программе нижнего уровня запускайте процесс приготовления порцию бетона. После

этого нажимайте на кнопки вызова VB-скриптов и следите за работой этих скриптов.



## Лабораторная работа № 5

### Изменение цвета задвижки с помощью VB-скрипта

Откройте редактор VB-скриптов и создайте процедуру SetZadvStatus(ZadvNo), которая предназначена для установки фонового цвета и режима мигания рисунка задвижки в зависимости от состояния задвижки.

Ниже приведен код этой процедуры:

```
Sub SetZadvStatus(ZadvNo)
    Dim ZadvName, Zadv

    Dim Open, Close, Opened, Closed

    ZadvName = "Zadv" & ZadvNo
    Set Zadv = ScreenItems(ZadvName)

    Open = HMIRuntime.Tags(ZadvName & ".Open").Read
    Opened = HMIRuntime.Tags(ZadvName & ".Opened").Read

    Close = HMIRuntime.Tags(ZadvName & ".Close").Read
    Closed = HMIRuntime.Tags(ZadvName & ".Closed").Read

    ' Задвижка открывается
    If Open = 1 Then
        Zadv.ForeColor = RGB(0,255, 0) ' Зеленый
        Zadv.BlinkMode = 2           ' мигает
    End If

    ' Задвижка открыта
    If Opened = 1 Then
        Zadv.ForeColor = RGB(0,255, 0) ' Зеленый
        Zadv.BlinkMode = 0           ' не мигает
    End If
```

' Задвижка закрывается

If Close = 1 Then

Zadv.ForeColor = RGB(255,255, 0) ' Желтый

Zadv.BlinkMode = 2 ' мигает

End If

' Задвижка закрыта

If Closed = 1 Then

Zadv.ForeColor = RGB(255,255, 0) ' Желтый

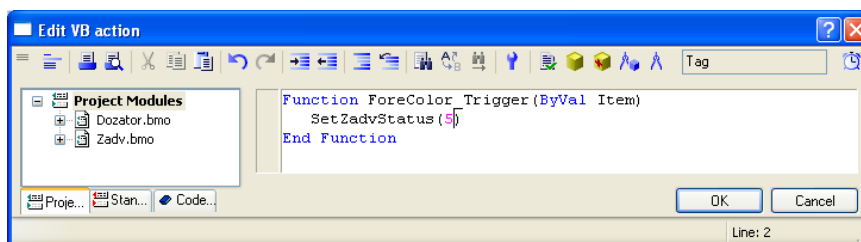
Zadv.BlinkMode = 0 ' не мигает

End If

End Sub

Как видно, в коде этой процедуры по номеру задвижки формируется имя задвижки и имя тега. Чтобы этот код работал правильно необходимо выполнить некоторые требования к именам задвижек и именам тегов. Например, для пятой задвижки рисунок задвижки должен иметь имя **Zadv5**, и тег для управления задвижкой должен иметь имя **Zadv5** типа **Zadv**.

Для свойства **ForeColor** рисунка задвижки необходимо определить динамику в виде VB-скрипта. Внутри этого скрипта следует вызывать процедуру **SetZadvStatus**. Например, для задвижки 5 окно определения VB-кода выглядит так:



Как видно, в этом случае имея одну процедуру, легко определить динамику для рисунка всех задвижек.

Запускайте проект Step 7 и WinCC одновременно в режим выполнения. В проекте Step 7 запускайте процесс приготовления порции бетона и следите за изменениями фонового цвета и режима мигания рисунка задвижки.

### Создание VB-скриптов в виде Actions

На рисунке дозатора разместите вертикальный прямоугольник высотой 100 пикселей. Пусть этот прямоугольник представляет собой датчик уровня материала, который меняет высоту, в зависимости от показаний датчика веса.



Для определения VB-скрипта типа **Action** в Редакторе VB-скрипта выберите вкладку **Actions**. С помощью команды **New/Action** контекстного меню узла **Actions** определите модуль **Level.bac**. Далее, также с помощью контекстного меню, определите процедуру типа **Action**. Теперь в правой части VB-редактора можно определить код этой процедуры.

**Option Explicit**

**Function** action

**Dim** Level, Rect, Screen

**Dim** Top, Height

Level = HMIRuntime.Tags("Dozat1.Wes").Read

**Set** Screen = HMIRuntime.Screens("BSU")

**Set** Rect = Screen.ScreenItems("Rect")

Top = 385


Top = Top - Level

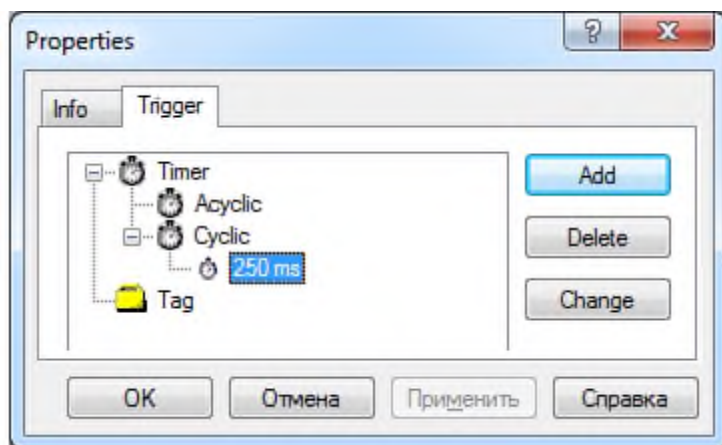
Rect.Top = Top

Rect.Height = Level

**End Function**

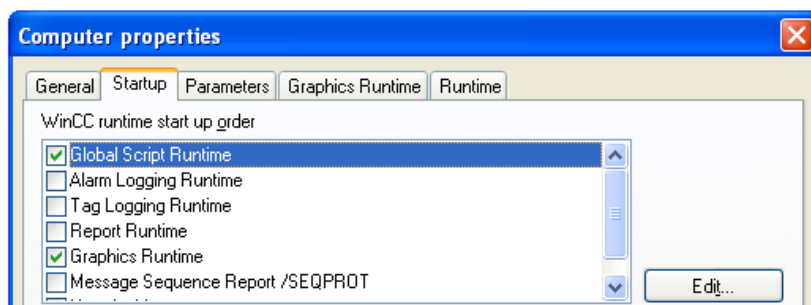
Как видно, в коде считывается значение тега Dozat1.Wes и потом его значение используется для расчета координаты Top и высоты Height прямоугольника.

Для определения триггера запуска Action щелкните на значке . При этом появляется следующее окно настройки триггера:



Нажатием кнопки **Add** в этом окне определите триггер, который вызывает скрипт типа Action, например, через каждые 250 мс.

Для того чтобы этот скрипт работал необходимо в окне свойств проекта установить флажок Global Script Runtime.



Запускайте проект Step 7 и WinCC одновременно в режим выполнения. В проекте Step 7 запускайте процесс приготовления порции бетона и следите за размером датчика уровня дозатора.

## Лабораторная работа № 6

### Синтаксис написания С-скриптов

Для изучения синтаксиса С-скриптов разместите на экране кнопок и для этих кнопок определите обработчик события мыши.

Параметры обработчика события в виде С-скрипта определяют ссылку на имя экрана, на имя графического объекта и его свойства. Чтобы проверить это напишите небольшой обработчик для события мыши кнопки:

```
#include "apdefap.h"

void OnClick(char* lpszPictureName,
             char* lpszObjectName,
             char* lpszPropertyName)
{
    printf ("Имя экрана = %s \r\n",
           lpszPictureName);
    printf ("Имя объекта = %s \r\n",
           lpszObjectName);
    printf ("Имя свойства= %s \r\n",
           lpszPropertyName);
}
```

Для выбора экрана процесса Main\_BSU.pdl в качестве главного экрана процесса в обработчике события мыши кнопки вызовите команду:

```
void OnClick(...)
{
    OpenPicture("Main_BSU.pdl");
}
```

Предположим, что на главном экране процесса BSU.pdl размещен объект типа Picture Window и дано ему название SubWnd. Для вывода вспомогательного экрана на этом объекте определите для кнопки следующий обработчик события мыши:

```
void OnClick(...)
```

```

{
    char *str;
    SetPictureName("BSU", "SubWnd", "Signal");

    str = GetPictureName("BSU", "SubWnd");
    printf ("Имя экрана = %s", str);
}

```

Для получения и установки свойств объектов экрана имеется GetProp- и SetProp-функции. В зависимости от типа свойства имеются четыре функции для получения и четыре функции для установки свойства.

Тип свойства	Считывание свойства	Установка свойства
Логический	<b>GetPropBOOL</b>	<b>SetPropBOOL</b>
Текстовой	<b>GetPropChar</b>	<b>SetPropBOOL</b>
Слово	<b>GetPropWord</b>	<b>SetPropBOOL</b>
Двойное слово	<b>GetPropDouble</b>	<b>SetPropBOOL</b>

Для примера разместите на экране BSU.pdl кнопку, для установки и чтения свойств текстового поля, которое предназначено для отображения показаний датчика веса дозатора. Для обработчика события мыши этой кнопки определите следующий код:

```

void OnClick(...)
{
    int i;
    long L;
    char *str;

    // Свойство Visible

```

```

SetPropBOOL("BSU.pdl", "Wes", "Visible", 1);
i = GetPropBOOL("BSU.pdl", "Wes",
                "Visible");
printf("Visible = %d \r\n", i);

// Свойство OutputValue
SetPropChar("BSU.pdl", "Wes",
            "OutputValue", "50");
str = GetPropChar("BSU.pdl",
                  "Wes", "OutputValue");
printf("OutputValue = %s \r\n", str);

// Свойство Left
SetPropWord("BSU.pdl", "Wes", "Left", 200);
i = GetPropWord("BSU.pdl", "Wes", "Left");
printf("Left = %i \r\n", i);

// Свойство BackColor
SetPropDouble("BSU.pdl", "Wes",
              "BackColor", 0xFF00FA);
L = GetPropDouble("BSU.pdl",
                  "Wes", "BackColor");
printf("BackColor = %X \r\n", L);
}

```

Для считывания и установки значений тегов язык С-скриптов имеет различных функций. Эти функции имеют одинаковые параметры, за исключением типов параметров. Ниже приведем некоторых из этих функций:

Тип тега	Считывание тега	Установка тега
Логический	<b>GetTagBit</b>	<b>SetTagBit</b>
Длиной байт	<b>GetTagByte</b>	<b>SetTagByte</b>
Текстовой	<b>GetTagChar</b>	<b>SetTagChar</b>
Слово	<b>GetTagWord</b>	<b>SetTagWord</b>
Двойное слово	<b>GetTagDWord</b>	<b>SetTagDWord</b>
Вещественный	<b>GetTagFloat</b>	<b>SetTagFloat</b>

Функции считывания значений тегов имеет один параметр – имя тега. Эти функции возвращают значение тега определенного типа. Функции установки значения тега имеют два параметра: имя тега и значение тега.

Для примера разместите кнопку на экране процесса и для него напишите обработчик события мыши:

```
void OnClick(...)
{
    int i;
    unsigned short w;

    // Ter Dozat1.Full
    SetTagBit("Dozat1.Full", 1);
    i = GetTagBit("Dozat1.Full");
    printf("Dozat1.Full = %d \r\n", i);

    // Ter Dozat1.Wes
    SetTagWord("Dozat1.Wes", 50);
    w = GetTagWord("Dozat1.Wes");
    printf("Dozat1.Wes = %d \r\n", w);
}
```

Запускайте проект WinCC в режим выполнения и нажимайте на кнопки вызова С-скриптов. Следите за работой этих скриптов.



### **Рекомендуемые источники**

1) SIMATIC HMI WinCC V7.2. Руководство пользователя. Нюрнберг.:  
Изд-во Siemens AG. 2012. – 1600 с.: ил: ISBN 5-06-005496-9. Электр. ресурс.  
<http://www.automation.siemens.com>